原创经典，程序员典藏

涉及众多热门技术，融汇6个开源项目，整合18种主流PHP解决方案
精选12个典型模块和6个PHP大型应用，实战PHP开发与建站



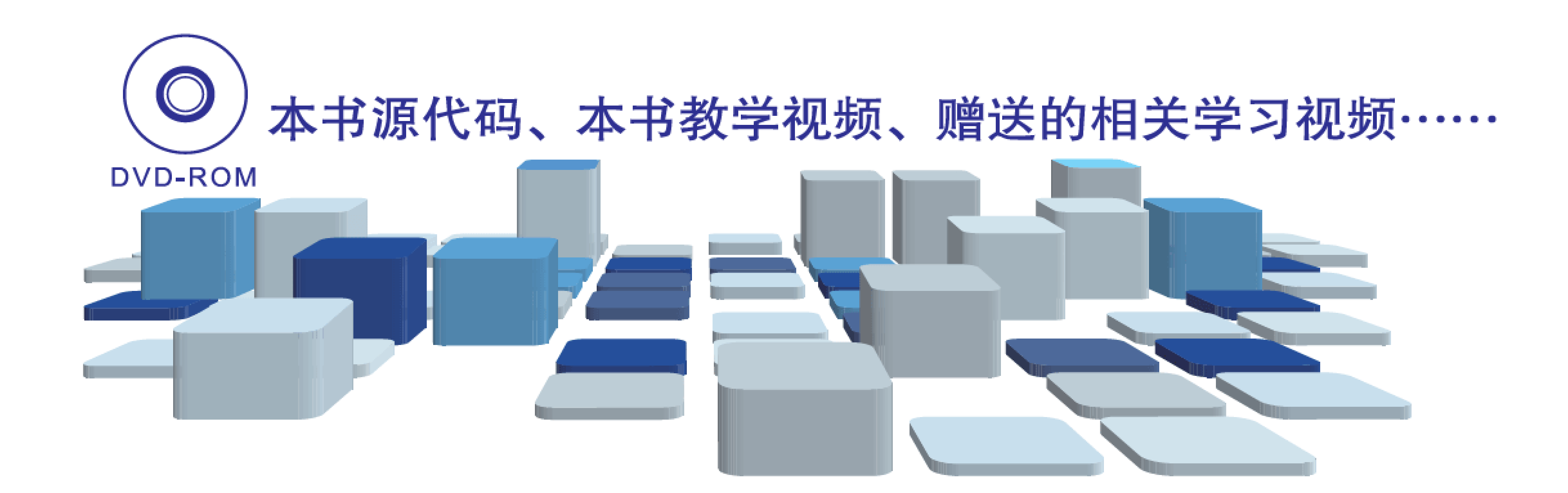
PHP典型模块与项目实战大全

（29小时多媒体教学视频）

杨宇　等编著

本书源代码、本书教学视频、赠送的相关学习视频……

DVD-ROM



清华大学出版社

PHP 典型模块与项目实战大全

杨　宇　等编著

清 华 大 学 出 版 社
北　京

内 容 简 介

本书以实战开发为原则，以 PHP 典型模块和项目开发为主线，通过 12 个高质量的 PHP 典型模块和 6 个 PHP 大型应用，向读者揭示了 Web 开发的整体结构，并详尽地介绍 PHP 开发与建站的技术要点。

本书附带 1 张 DVD，内容是作者为本书录制的全程多媒体语音教学视频及本书所涉及的源代码。

本书共 22 章，分为 3 篇。第 1 篇是 PHP 开发基础，涵盖的内容有 Web 开发基础、开发环境的安装与配置、PHP 语言参考、PHP 简单实例；第 2 篇为典型模块开发与应用，涵盖的内容有留言板模块、注册及登录验证模块、上传与下载模块、内容采集器模块、日程表模块、聊天室程序、计费系统模块、充值平台模块、在线调查模块、WAP 资讯模块、广告管理模块、文件管理器模块；第 3 篇为项目开发实战，涵盖的内容有内容管理系统、娱乐联播软件、宝藏游戏、图片交易平台、在线购物平台、Joomla 扩展开发与快速建站。

本书内容全面、深入，既是优秀的学习教程，也可用作参考手册。书中的大量实例和深入分析均来自于作者多年的实际工作经验，对于解决开发人员在实际开发中所面临的各种问题有重要的借鉴意义。本书适合各层次的 PHP 程序员、爱好者和 Web 开发人员阅读。

图书在版编目（CIP）数据

PHP 典型模块与项目实战大全 / 杨宇等编著. —北京：清华大学出版社，2011.8
ISBN 978-7-302-25821-6

Ⅰ. ①P…　Ⅱ. ①杨…　Ⅲ. ①PHP 语言 – 程序设计　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2011）第 113551 号

责任编辑： 夏兆彦
责任校对： 徐俊伟
责任印制：
出版发行： 清华大学出版社　　**地　　址：** 北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编：** 100084
社 总 机： 010-62770175　　**邮　　购：** 010-62786544
投稿与读者服务： 010-62795954，jsjjc@tup.tsinghua.edu.cn
质量反馈： 010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销： 全国新华书店
开　　本： 185×260　**印　　张：** 41.5　**字　　数：** 1036 千字
（附光盘 1 张）
版　　次： 2011 年 8 月第 1 版　　**印　　次：** 2011 年 8 月第 1 次印刷
印　　数： 1～ 000
定　　价： 　元

产品编号：040086-01

前　言

为什么要写这本书

PHP 语言已经流行多年，并深入到 Web 开发的各个领域。它能应用于 Windows、Linux 和 Mac OS 等平台上。基于 PHP 的开发已经广泛地被互联网公司采用，它已经变得越来越重要。但是 Web 开发往往需要结合多种技术进行，因此具有较高的“门槛”，这使很多 Web 开发爱好者受知识结构的限制，技术上难以提高。

本书结合 PHP 开发中的各种流行技术和典型应用，介绍 PHP 实际开发中涉及的最常用、最典型的一些开发模块和项目开发案例。帮助读者理解 Web 开发的特点和 PHP 开发涉及的各种热门技术与主流框架，并熟悉各种应用的解决方案，使读者在技术上有质的突破，从而提高实际开发水平和项目实战能力。本书中的大量实例和深入的分析均来自于作者多年的工作经验，对于解决开发者所面临的各种问题有重要的借鉴意义。

本书有何特色

1．循序渐进，由浅入深

本书从 Web 开发的基本概念讲起，并结合开发环境的搭建和 PHP 语言基础，为各阶段的读者顺利进入 Web 开发领域奠定了基础。然后从易到难安排了 12 个典型应用模块和 6 个项目案例开发，以提高读者的实战能力。

2．注重原理，结合实际分析

本书注重原理讲解，每个模块都从基础设计和系统需求讲起，针对每个模块的核心技术原理，作者都给出了原理图帮助读者理解，并在后续的实际开发中将该原理应用到实际项目中。

3．构架剖析，源代码精解

随着 OOP 开发模式在编程方面的广泛应用，基于各种架构的开发已经成为当前程序开发的主流。本书也顺应这种趋势，在模块实现的过程中应用了多种当前流行的架构，并对每个涉及的架构都做了深入的剖析并从原理上进行了阐释。最后通过对源代码的详细讲解，使读者可以清楚、直观地掌握该架构。

4．模块设计，独立应用，便于扩展

本书给出了 PHP 开发的 12 个典型应用模块，这些模块涵盖 PHP 开发的热门技术和主

流应用。这些模块独立成章，每章内容都明确地阐释某个主题或某类典型应用。作者在具体内容的组织上，注意为读者预留扩展的接口并做了详细的说明，便于读者扩展该模块。

5．经典模块结合主流应用，适合各个层次的读者

本书讲解模块开发和项目开发时，采用了经典应用实例和当前主流技术相结合的方式，使得内容新颖，又能很好地描述技术问题，也使得各个层次的读者在阅读本书后均有所获益。

6．项目案例典型，实战性强，有较高的应用价值

本书提供了 6 个项目实战案例，这些案例分别使用不同的技术组合实现，便于读者融会贯通地理解本书中所介绍的技术。这些案例稍加修改，便可用于实际的项目开发中。

7．附带多媒体语音教学视频，提高学习效率

为了便于读者理解本书内容，提高学习效率，作者专门为本书内容录制了大量的多媒体语音教学视频。这些视频和本书涉及的源代码一起收录于配书光盘中。

8．提供完善的技术支持和售后服务

本书提供了专门的技术支持邮箱：bookservice2008@163.com。读者在阅读本书过程中有任何疑问都可以通过该邮箱获得帮助。

本书内容及知识体系

第 1 篇　PHP 开发必备基础（第 1～4 章）

本篇介绍 Web 开发的基础知识、开发环境的安装与配置、PHP 语言参考和 PHP 的入门实例，这些构成了 Web 开发的基础内容。

第 2 篇　PHP 典型模块开发与应用（第 5～16 章）

本篇介绍当前 Web 开发中的 12 个典型应用，笔者结合实例阐述技术原理。该篇涉及的内容广泛，包括留言板模块、注册及登录验证模块、上传与下载模块、内容采集器模块、日程表模块、聊天室程序、计费系统模块、充值平台模块、在线调查模块、WAP 资讯模块、广告管理模块、文件管理器模块等。本篇在内容设计上采用难度逐步递增的策略，使得读者的应用能力随着学习的深入而逐步提高。

第 3 篇　PHP 项目开发实战（第 17～22 章）

本篇主要介绍 6 个 PHP 的大型应用，包括内容管理系统、娱乐联播软件、宝藏游戏、图片交易平台、在线购物平台和 Joomla 快速建站。本篇内容旨在提高读者整体项目的设计能力以及框架的理解与应用能力。

配书光盘内容介绍

为了方便读者阅读本书，本书附带 1 张 DVD 光盘。内容如下：

- 本书所有实例的源代码；
- 本书每章内容的多媒体教学视频；
- 免费赠送的 25 小时 MySQL 入门教学视频。

适合阅读本书的读者

- Web 开发爱好者；
- PHP 程序开发人员；
- 希望提高项目开发水平的人员；
- 专业培训机构的学员；
- 软件开发项目经理；
- 中小企业网站维护人员；
- 站长或网站的架构设计与管理人员。

阅读本书的建议

- 本书的设计思路是由浅入深，难度逐步加大，因此建议读者从本书的第 1 章开始学习，这样知识的跨度不会太大，便于对知识消化和理解。
- 对有一定基础的读者也建议从头读起，因为本书对很多技术细节和原理做了详尽的阐述。有 Web 开发基础的读者，通过这种循序渐进的学习可以获得更好的效果，同时对具体的技术问题会有更深入的认识。
- 可以先对书中的模块和项目案例阅读一遍，然后结合光盘中提供的多媒体教学视频再理解一遍，这样理解起来就更加容易，也会更加深刻。

本书作者及编委会成员

本书由杨宇主笔编写。其他参与编写和资料整理的人员有武冬、郅晓娜、孙美芹、卫丽行、尹翠翠、蔡继文、陈晓宇、迟剑、邓薇、郭利魁、金贞姬、李敬才、李萍、刘敬、陈慧、刘艳飞、吕博、全哲、佘勇、宋学江、王浩、王康、王楠、杨宗芳、张严虎、周玉、张平、张靖波、周芳、杨罡、于海滨、张晶杰、张利峰、杨景凤、陈锴、郑剑锋、叶佩思、张涛、赵东彪、王双。在此一并表示感谢！

本书编委会成员有欧振旭、陈杰、陈冠军、项宇峰、张帆、陈刚、程彩红、毛红娟、聂庆亮、王志娟、武文娟、颜盟盟、姚志娟、尹继平、张昆、张薛。

目　　录

第 1 篇　PHP 开发必备基础

第 3 篇　PHP 项目开发实战

第 1 篇　PHP 开发必备基础

第 1 章　Web 开发基础

Web 开发涉及的技术和知识非常广泛。本章将由浅入深、循序渐进地向读者介绍 Web 服务器原理、PHP 发展、HTML 的基础知识。在 JavaScript 部分会结合几个简单的例子，讲解客户端脚本语言的基本原理和具体应用。

PHP 作为当前被广泛应用的服务器端脚本语言，已经帮助许多互联网公司构建了完善的业务模型和应用程序。本章将回顾 PHP 的发展历史，向读者介绍 Web 开发相关的基础知识，为下面的学习打下良好的基础。

本章主要涉及的知识点如下。

- PHP 主要应用领域：服务器端脚本、命令行脚本、编写桌面应用。
- HTML 的结构：头部（Head）和主体（Body）两大部分。
- JavaScript 在 HTML 中的 3 种引用方式：内部引用、外部引用、内联引用。

1.1　PHP 简介

PHP（Hypertext Preprocessor，超级文本预处理语言），它是一种被广泛使用的开放源代码的多用途脚本语言。PHP 语言的风格类似于 C 语言，初学者容易掌握。而且已经在 Web 领域被广泛的运用。由于 PHP 具有“易用性”的特点，因此它可以帮助 Web 开发人员快速地搭建需要完成的应用。

1.1.1　PHP 有什么特点

PHP 的特点可以概括为以下几个方面。

- 开源性：源码开放，开发者可以共同参与到众多的 PHP 开源项目中。
- 免费的 PHP：任何人都可以不受任何限制地使用 PHP 源代码。在中国，很多互联网公司都采用 LAMP（Linux、Apache、MySQL、PHP）平台搭建自己的网站，这种组合模式兼顾了低投入（LAMP 整套的建站解决方案，降低企业的投入成本）和高性能，同时还具有良好的安全性。
- PHP 的易用性：PHP 较其他脚本语言学习成本更低。由于语法接近 C 语言，使得初学者更容易掌握。
- 执行效率高：PHP 消耗较少的系统资源就可以获得很高的执行效率。
- 图像处理：PHP 提供了丰富的图像函数库。这些函数库用于动态创建图像，并可以直接将图像流输出到浏览器。
- 面向对象：PHP 早期版本的代码在本质上是限于面向过程的，过程代码通过允许

过程之间的调用来实现重用。在 PHP 的新版中，面向对象有了本质上的改变，现在 PHP 完全可以用来开发大型企业级应用。PHP 相对于其他脚本语言而言，学习成本低、实用性强、执行效率较高，可以胜任大型应用。

1.1.2　PHP 能做什么

PHP 初学者的一个主要疑问就是：PHP 能用来做什么？按照 PHP 手册上的经典解释就是：PHP 能做任何事！

下面为读者具体的介绍 PHP 主要的应用领域。PHP 是运行于服务端的脚本程序，因此可以用其完成任何 CGI（Common Gateway Interface）程序能够完成的工作。CGI 工作流程如图 1.1 所示。CGI 主要用于完成在 B/S（Browser/Server）模式下的应用开发，例如，动态页面输出、获取表单数据、创建或接收 Session/Cookies。

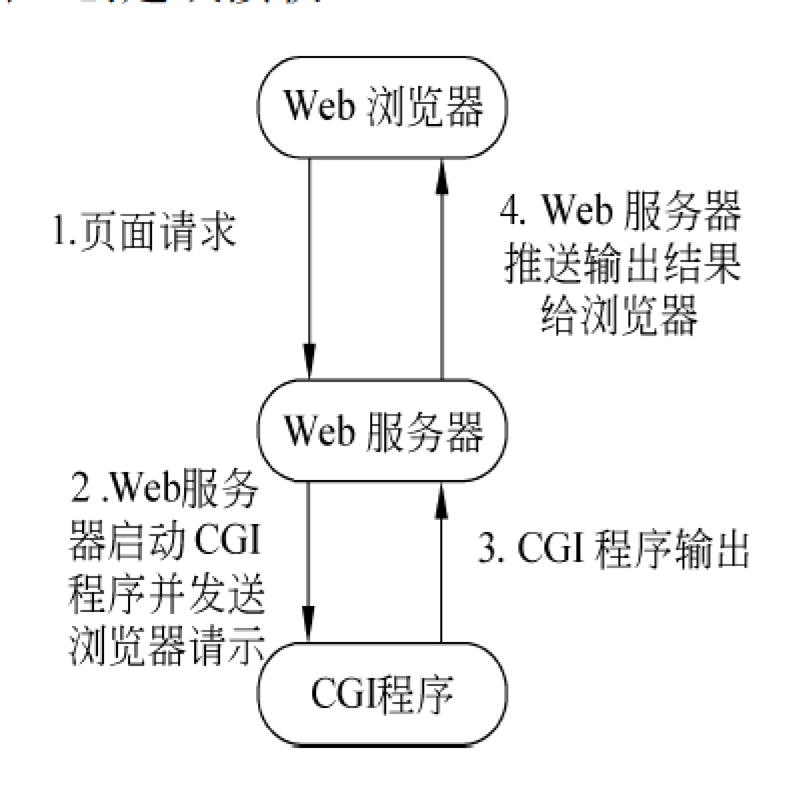


图 1.1　CGI 工作流程示意图

PHP 主要应用于以下 3 个领域。

1. 服务端脚本

服务器脚本运行模式需要具备 3 个条件：PHP 解析器（CGI 或者服务器模块）、Web 服务器、Web 浏览器。

具体执行流程：首先运行 Web 服务器，然后安装并配置 PHP，最后可以用 Web 浏览器访问 PHP 程序的输出，即浏览服务端的 PHP 页面（或数据输出）。

2. 命令行脚本

通过命令行模式运行 PHP 脚本，这种模式下不需要服务器的支持或者依赖浏览器的触发，在命令行脚本模式下仅需要 PHP 解析器来执行。通常这种方法被用在 Windows 和 Linux 平台下做日常运行脚本使用，如某些守护程序等。当然这些脚本也可以用来处理简单的文本。

3. 编写桌面应用程序

PHP 也许并不完全适合用来做图形界面应用程序，但是开发人员同样可以在客户端程序应用某些 PHP 的高级特性，通过使用 PHP-GTK 来编写这些程序。PHP-GTK 是 PHP 的

一个扩展，并不包含在标准的开发包中，开发人员需要单独编译它。通过 PHP-GTK 可以建立独立的 GUI（用户图形界面）应用程序，例如，音频播放器、文本编辑器、数据库后台等。

1.1.3　用什么技术来搭建 PHP 环境

目前，所有的主流操作系统上 PHP 都能够运行，包括 Linux、Unix 系列、Microsoft Windows、Mac OS X、RISC OS 等。新版本的 PHP 支持大多数的 WebServer，包括 Apache、Nginx Microsoft Internet Information Server（IIS）、Personal web Server（PWS）、Netscape、iPlant server、Oreilly Website Pro Server、Caudium、Xitami、OmniHTTPd 等。

当用 PHP 来做 Web 开发时，可以使用 Linux 系列的操作系统或 Windows NT 操作系统平台。Web 开发人员通常会选择系统稳定性更好的 Linux 系列系统，作为 PHP 的执行平台，实际的使用经验证明 Linux 系列操作系统拥有更好的系统稳定性。

在 Linux 系列操作系统下通常是 LAMP（Linux、Apache、MySQL、PHP）的组合模式，许多互联网公司都以此作为 Web 开发框架来搭建自身的应用，并获得了成功。同时，由于 Linux、Apache、MySQL、PHP 都是可以免费使用的，因此搭建开发框架的成本也相对较低。

LAMP 是一个缩写，它是指一组通常被一起使用来搭建 Web 服务的自由软件，如下所示。

- Linux：服务器操作系统。
- Apache：开源的 Web 服务器软件。
- MySQL：高效的关系型数据库。
- PHP（Perl 或 Python）：服务器脚本语言。

Apache 是 LAMP 架构最核心的部件（Web Server）。Apache 具有开源、稳定、模块丰富的优势。但是，不如一些轻量级的 Web 服务器高效（如 Nginx），轻量级的 Web 服务器对于静态文件的响应能力来说远高于 Apache 服务器。在这里简要的介绍 Nginx 服务器软件。Nginx 是一个高性能的 HTTP 和反向代理服务器，Nginx 以它的稳定性、丰富的功能集、示例配置文件和较低的系统资源消耗而闻名。

MySQL 数据库是多线程、支持多用户的 SQL 数据库服务器。它的性能和稳定性都十分优秀，同时也提供了丰富的管理功能。针对高并发访问和频繁读写操作，通常可以使用主从结构或数据库集群、优化表结构的方式解决。

当然，PHP 可以支持众多主流的数据，它几乎涵盖了目前全部的主流数据库，PHP 可以支持的数据库如表 1.1 所示。PHP 还支持很多不太流行的小众数据库，这里不再一一列出。

表 1.1　PHP支持数据库列表

数据库名称	描　述
Adabas D	是一种新的关系型数据库
dBase	第一个在微型计算机上被广泛使用的数据库管理系统

续表

数据库名称	描　述
Empress	Empress嵌入式实时数据库是基于知识和规则的关系型的稳定可靠的数据库系统，具有免维护、适应性强、模块化、全分布、多平台、易裁剪、可扩展和开放性强的优点
IBM DB2	IBM公司研制的一种关系型数据库系统。DB2主要应用于大型系统，具有较好的可伸缩性，可支持从大型机到单用户环境，应用于OS/2、Windows等平台下
Informix	是IBM公司出品的关系数据库管理系统（RDBMS）家族。作为一个集成解决方案，它被定位为作为IBM在线事务处理（OLTP）旗舰级数据服务系统
InterBase	是一种关系数据管理系统，它提供了在单机或多用户环境中的快速数据处理及共享的工具。InterBase的核心是提供透明的多机种支持的网络运行服务器技术。InterBase是可以在Windows 95、Windows NT、Novell NetWare及多种UNIX操作系统上运行的工具
mSQL	mSQL（mini SQL）是一个单用户数据库管理系统。由于它的小巧精悍，使其开发的应用系统受到互联网用户青睐。mSQL是一种小型的关系数据库，性能平平，对SQL语言的支持也不够完全，但可以满足一些网络数据库应用。由于mSQL较简单，在运行简单的SQL语句时速度比MySQL略快，而MySQL在线程和索引上下了功夫，运行复杂的SQL语句时比mSQL、PostgreSQL等都要快一些
Direct MS-SQL	MS-SQL Server 是 Microsoft 公司推出的Windows 平台上最为流行的中型的数据库管理系统。特点：采用客户/服务器体系结构图形化的用户界面，使系统的管理更加直观和简单，与Windows操作系统的有机集成，多线程体系结构设计，提高了系统对用户并发访问的速度
MySQL	是一个快速、多线程、多用户和强壮的SQL数据库服务器，由于其体积小、速度快、总体拥有成本低，尤其是开放源码的特点便于二次开发和定制服务，因此许多网络公司选择MySQL作为数据库
ODBC	是微软公司开放服务结构中有关数据库的一个组成部分，它建立了一组规范，并提供了一组对数据库访问的标准API（应用程序编程接口）。这些API利用SQL来完成其大部分任务，ODBC本身也提供了对SQL语言的支持，用户可以直接将SQL语句送给ODBC
Oracle	是世界领先的信息管理软件开发商，因其复杂的关系数据库产品而闻名。Oracle的关系数据库是世界第一个支持SQL语言的数据库，许多大型网站都选用了Oracle系统
PostgreSQL	是以加州大学伯克利分校计算机系开发的POSTGRES版本4.2为基础的对象关系型数据库管理系统
SQLite	是一款轻型的数据库，是遵守ACID的关联式数据库管理系统。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常得低
Sybase	美国Sybase公司研制的一种关系型数据库系统，是一种典型的UNIX或Windows NT平台上客户机/服务器环境下的大型数据库系统

除了使用上述数据库以外，PHP 还提供一个 DBX 扩展库，用户可以自由地使用该扩展库支持的任何数据库。此外，PHP 具有一个丰富的 ODBC 数据库连接函数库（Open Database Connection Standard，开放数据库连接标准），因此可以连接任何其他支持该标准的数据库。由于 PHP 在数据库方面优秀的兼容性，它可以在不同数据库之间切换，这进一步扩大了应用范围。

说明：本书所使用的程序内容及范例程序，大部分都是基于 Linux+Apache+MySQL+PHP 这个开发框架的。

1.2　HTML 基础

HTML（HyperText Mark-up Language，超文本标记语言或超文本链接标示语言），是目前网络上应用最为广泛的语言，也是构成网页文档的主要语言。HTML 文本是由 HTML 命令组成的描述性文本，HTML 命令可以说明文字、图形、动画、声音、表格、链接等。HTML 的结构包括头部（Head）、主体（Body）两大部分，其中头部描述浏览器所需的信息，而主体则包含所要说明的具体内容。

1.2.1　HTML 文档基本结构

在 HTML 网页文档的基本结构中主要包含以下几种标记。

1．HTML 文件标记

<HTML>和</HTML>标记放在网页文档的最外层，表示这对标记间的内容是 HTML 文档。<HTML>放在文件开头，</HTML>放在文件结尾，在这两个标记中间嵌套其他标记。

2．HEAD 文件头部标记

文件头用<HEAD>和</HEAD>标记，该标记出现在文件的起始部分。标记内的内容不在浏览器中显示，主要用来说明文件的有关信息，如文件标题、作者、编写时间、搜索引擎可用的关键词等。

在 HEAD 标记内最常用的标记是网页主题标记，即 TITLE 标记，它的格式为：

```
<TITLE>网页标题</TITLE>
```

网页标题是提示网页内容和功能的文字，它将出现在浏览器的标题栏中。一个网页只能有一个标题，并且只能出现在文件的头部。

3．BODY 文件主体标记

文件主体用<BODY>和</BODY>标记，它是 HTML 文档的主体部分。网页正文中的所有内容包括文字、表格、图像、声音和动画等都包含在这对标记对之间。通常的标记样式如下：

```
<BODY background="image-URL" bgcolor="color" text="color" link="color"
alink="color" vlink="color" leftmargin="value" topmargin="value"> …
</BODY>。
```

1.2.2　简单的 HTML 示例

了解了 HTML 的基本结构和组成，下面来创建第一个 HTML 文档。先建立一个空白

TXT 文档，在其中书写如下代码，完成后将文件名保存为 Hello.html。

```
<HTML>
<HEAD>
<TITLE>一个简单的 HTML 示例</TITLE>
</HEAD>
<BODY>
<CENTER>
  <H1>Hello World</H1>
  <BR>
  <HR>
  <FONT SIZE= 7 COLOR= red> 欢迎光临 </FONT>
</CENTER>
</BODY>
</HTML>
```

下面逐一讲解代码中用到的标记：

- <HTML></HTML>中间的部分都是 HTML 元素，它表示该文档是用超文本标识语言（HTML）编写的。事实上，现在常用的 Web 浏览器都可以自动识别 HTML 文档，并不要求有此标记，也不对它进行任何操作。但是为了使 HTML 文档能够适应不断变化的 Web 浏览器，还是应该养成不省略这对标记的良好习惯。
- <HEAD></HEAD>是 HTML 文档的头部标记，在浏览器窗口中，头部信息是不被显示在正文中的。在此标记中可以插入其他标记，用以说明文件的标题和整个文件的一些公共属性。若不需头部信息则可省略此标记，但是笔者不建议省略它。
- <TITLE>和</TITLE>嵌套在<HEAD>头部标记中。标记之间的文本是文档标题，它被显示在浏览器窗口的标题栏。
- <BODY></BODY>标记一般不省略，标记之间的文本是正文，是在浏览器中要显示的页面内容。

上面的这几对标记在文档中都是唯一的，HEAD 标记和 BODY 标记都嵌套在 HTML 标记中。

1.2.3　创建网页上的列表

列表是网页最基本的形式，同时也是最重要的组成，制作列表所用到的标签如下。

1. <table></table>标记

<table></table>“标记对”用来创建一个表格，属性见表 1.2。

表 1.2　Table属性对照表

属　　性	描　　述
<table bgcolor="">	设置表格的背景色
<table border=" ">	设置边框的宽度，若不设置此属性，则边框宽度默认为0
<table bordercolor=" ">	设置边框的颜色
<table bordercolorlight="">	设置边框明亮部分的颜色（当border的值大于等于1时才有用）
<table bordercolordark="">	设置边框昏暗部分的颜色（当border的值大于等于1时才有用）
<table cellspacing="">	设置表格格子之间空间的大小
<table cellpadding="">	设置表格格子边框与其内部内容之间空间的大小
<table width="">	设置表格的宽度，单位用绝对像素值或总宽度的百分比

说明：以上各个属性可以结合使用。有关宽度、大小的单位用绝对像素值。有关颜色的属性使用十六进制 RGB 颜色码或 HTML 语言给定的颜色常量名（如 Silver 为银色）。

2．<tr></tr><td></td>标记

<tr></tr>“标记对”用来创建表格中的每一行，此“标记对”只能放在<table></table>“标记对”之间使用。在此“标记对”之间加入文本将是无效的，因为在<tr></tr>之间只能紧跟<td></td>“标记对”才是有效的语法。<td></td>“标记对”用来创建表格中一行中的每一个格子，此“标记对”也只有放在<tr></tr>“标记对”之间才是有效的，假如想要输入文本需要放在<td></td>“标记对”中才有效（能够被浏览器正确识别并显示出来）。<table></table>、<tr></tr>和<td></td>“标记对”的关系如下所示：

```
<table>
<!--创建一行-->
<tr>
<!--创建一个格子（这里总共创建了三个格子）-->
<td>要输出的文本只能放在此处</td>
<td>要输出的文本只能放在此处</td>
<td>要输出的文本只能放在此处</td>
</tr>
<!--最外层-->
</table>
```

此外，<tr>标记还有 align 和 valign 属性，其中 align 是水平对齐方式，取值为 left（左对齐）、center（居中）、right（右对齐）；而 valign 是垂直对齐方式，取值为 top（靠顶端对齐）、middle（居中间对齐）或 bottom（靠底部对齐）。<td>标记具有 width、colspan、rowspan 和 nowrap 属性。Width 属性是格子的宽度，单位用绝对像素值或总宽度的百分比。colspan 属性设置一个表格格子跨占的列数（默认值为 1）。rowspan 属性设置一个表格格子跨占的行数（默认值为 1）。nowrap 属性禁止表格格子内的内容自动断行。

3．<th></th>标记

<th></th>“标记对”用来设置表格头，通常是黑体居中文字。下面的代码创建了一个 3 行 4 列的表格，代码比较简单，这里不做过多说明。

```
<html>
<head>
<title>表格标记的综合示例</title>
</head>
<body>
<table   border="1"   width="80%"   bgcolor="#E8E8E8"   cellpadding="2"
bordercolor="#0000FF"
bordercolorlight="#7D7DFF" bordercolordark="#0000A0">
  <tr>
    <th width="33%" colspan="2" valign="bottom">意大利</th>
    <th width="36%" colspan="2" valign="bottom">英格兰</th>
    <th width="36%" colspan="2" valign="bottom">西班牙</th>
  </tr>
  <tr>
    <td width="16%" align="center">AC 米兰</td>
```

```
    <td width="16%" align="center">佛罗伦萨</td>
    <td width="17%" align="center">曼联</td>
    <td width="17%" align="center">纽卡斯尔</td>
    <td width="17%" align="center">巴塞罗那</td>
    <td width="17%" align="center">皇家社会</td>
  </tr>
  <tr>
    <td width="16%" align="center">尤文图斯</td>
    <td width="16%" align="center">桑普多利亚</td>
    <td width="17%" align="center">利物浦</td>
    <td width="17%" align="center">阿申纳</td>
    <td width="17%" align="center">皇家马德里</td>
    <td width="17%" align="center">…</td>
  </tr>
  <tr>
    <td width="16%" align="center">拉齐奥</td>
    <td width="16%" align="center">国际米兰</td>
    <td width="17%" align="center">切尔西</td>
    <td width="17%" align="center">米德尔斯堡</td>
    <td width="17%" align="center">马德里竞技</td>
    <td width="17%" align="center">…</td>
  </tr>
</table>
</body>
</html>
```

1.2.4　创建图像和链接

HTML 用<a>来表示链接。<a>可以指向任何一个文件源，如一个 HTML 网页、一个图片、一个影视文件、一个 E-mail 地址等。例如，下面代码表示链接显示的文字。

```
<a href="http://www.google.com">google</a>
```

链接的属性如表 1.3 所示。

表 1.3　链接属性表

属　性	说　明
<target>	可以在一个新窗口打开链接 target=_blank
<title>	可以让鼠标悬停在链接上时显示相关文字备注
<name>	可跳转到文件的指定部位。要设置一对，一个是name，另一个是设定一个href指向这个name

在 HTML 中，用<img>在插入图片。例如，<img src="url">，其中 url 可以是图片的路径和文件名，也可以是相对的路径。图像标记的属性如表 1.4 所示。

表 1.4　图像标记属性表

属　性	说　明
<alt>	图片说明，便于搜索引擎找到网页
<align>	改变图片的垂直对齐方式和水平对齐方式
<height>or<width>	改变图片大小

下面的示例展示了链接标记、图像标记在 HTML 中的具体应用，代码如下：

```
<html>
<body>
<p>
可以将图片作为一个链接，点击这个图片
<a href="../../.html"><imgsrc="../images/smile.jpg" ></a>
</p>
</body>
</html>
```

1.2.5　创建表单

表单（Form）是 HTML 最常见的标记之一，用于采集和提交用户输入的信息。Form 表单由 3 个要素构成，分别是表单控件、Action 和 Method。

- 表单控件（Form Controls）：通过各种控件，用户可以输入文字信息、可以选择内容。表单中常见的控件如表 1.5 所示。
- Action：指明处理表单信息的文件。
- Method：表示发送信息的方式。Method 有 Get 和 Post 两个值。Get 是将表单控件的 name 和 value 信息编码后通过 URL 发送，通常是为了取得和显示数据；Post 则通过 HTTP 发送，用于数据的保存和更新。

表 1.5　表单常用控件说明表

表单控件（Form Contros）	说　明
input type="text"	单行文本输入框
input type="submit"	将表单（Form）里的信息提交给表单里 action 所指向的文件
input type="checkbox"	复选框
input type="radio"	单选框
select	下拉框
textArea	多行文本输入框
input type="password"	密码输入框（输入的文字用*表示）

表单控件常见的有单行输入文本框、复选框、单选框、下拉框和多行输入框。下面作具体的介绍。

1．单行输入文本框

单行输入文本框控件允许用户输入一行文本。代码示例如下：

```
<form action="http://../yourname.asp" method="get">
<h5>请输入你的姓名：<h5>
<input type="text" name="yourname">
<h5>请输入你的密码：<h5>
<input type="password" name="yourpw">
<input type="submit" value="提交">
</form>
```

【代码解读】

上面示例中，“密码”输入框用标记（input type="password"）表示，“提交”按钮用

标记（input type="submit"）表示。

2．复选框（checkbox）

复选框控件允许用户选中多个选项。代码示例如下：

```
<input type="checkbox" name="sports" value="snooker" checked>斯诺克<br>
<input type="checkbox" name="sports" value="tennis">网球<br>
<input type="checkbox" name="sports" value="basketball">篮球<br>
```

注意：checked表示默认值，是已选中状态。

3．单选框（radio）

单选框控件允许用户在提供的多个选择中，选择一个选项。代码示例如下：

```
<input type="radio" name="sports" value="snooker" checked>斯诺克<br>
<input type="radio" name="sports" value="tennis">网球<br>
<input type="radio" name="sports" value="basketball">篮球<br>
```

4．下拉框（select）

下拉框控件允许用户选择一个下拉菜单中的选项。代码示例如下：

```
<select  name="sports">
<option value="snooker">斯诺克
<option value="tennis">网球
<option value="basketball">篮球
</select>
```

5．多行输入框（textarea）

多行输入框控件允许用户输入多行文本。代码示例如下：

```
<textarea name="suggest" cols ="50" rows = "4"></textarea>
```

注意：cols 为文本框宽度，rows 为文本框高度。

1.3　JavaScript 基础

JavaScript 是一种高安全性的脚本语言，它基于对象（Object）和事件驱动（Event Driven）。JavaScript 作为动态网页制作的编程语言，被广泛地应用于 Web 页面处理页面逻辑中。在 HTML 基础上，使用 JavaScript 可以开发交互式 Web 网页。JavaScript 的出现，使得网页和用户之间实现了一种实时性的、动态的、交互性的关系，使网页包含更多活跃的元素和更加精彩的内容。JavaScript 语言本身十分简练，又是在客户端执行，因此大大提高了网页的浏览速度和交互能力。

1.3.1　JavaScript 在 HTML 中的嵌入方式

JavaScript 作为客户端的脚本语言，主要的用途就是帮助 HTML 处理部分交互逻辑。

JavaScript 语言同其他脚本语言一样，通过控制语句、函数、对象方法属性等来实现编程。先来介绍 JavaScript 是如何同 HTML 交互的，也就是嵌入的方式。根据 JavaScript 在 HTML 中加载方式的不同，嵌入方式可以分为内部引用、外部引用和内联引用。

1．内部引用

通过<Script></Script>标记嵌入 JavaScript，这是最常用也是最简便的一种引用方式，可以在 HTML 代码的任何位置嵌入。

```
<head>
    <script type="text/javascript">
    document.write ("Hello world!") ;
    </script>
</head>
```

2．外部引用

通过引用 HTML 文件的方式加载 JavaScript 文件，这种方式可以使 HTML 代码更简洁，方便代码复用。

```
<head>
    <script type="text/javascript" src=ga.js"></script>
</head>
```

标准加载方法是把 JavaScript 文件放到 head 标记内，但对于一些非关键的 JavaScript 文件，可以放到 HTML 文件的底部（如一些流量统计代码）。这样做可以提高网页访问速度，获得更好的用户体验，代码位置如下：

```
    <script type="text/javascript" src="ga.js"></script>
    </body>
</html>
```

3．内联引用

通过 HTML 标记的触发事件属性实现，比如通过 onclick 事件直接调用 JavaScript 代码。在 HTML 中有很多这样的事件属性，通常都是配合 JavaScript 这样的前端脚本语言来使用，示例代码如下：

```
<input type="button" value="" onclick="alert ('你内联引用的方式调用了JavaScript 代码') ;">
```

【代码解读】

以上代码会在页面生成一个按钮标记，当单击该按钮时会调用 input 标记的 onclick 属性，弹出一个提示框。

在实际应用中，以上 3 种方法通常会根据不同需求组合使用。

1.3.2　控制 IE 的弹出页面

最基本的弹出窗口是通过 window.open()方法来实现的。读者对下面这段代码应该不陌生，执行该方法将会在浏览器中打开一个新的 index.html 页面。

```
<script LANGUAGE="javascript">
window.open ('index.html')
</script>
```

1.3.1 节提到过 JavaScript 代码在 HTML 中的嵌入形式，所以应该放在<script LANGUAGE ="javascript">标记和</script>之间。window.open ('index.html') 的作用是打开一个新窗口，并装载 URL 指定的文档，如果没指定 URL 将会装载一个空白文档。

注意：如果载入的文件与主窗口不在同一路径下，前面应写明路径，绝对路径（http://）或相对路径（../）均可，可以用单引号也可以用双引号，只是要注意符号统一。

1. 通过改变 window.open()的参数来弹出窗口

通过改变 window.open()函数中的参数，可以对弹出窗口的外观进行具体的设置，例如：弹出窗口的外观、尺寸大小、弹出位置等。

```
<script LANGUAGE="javascript">
<!--
window.open('index.html','newwindow','height=100,width=400,top=0,left=0,to
olbar=no,menubar=no,scrollbars=no, resizable=no,location=no,status=no')
//注意写在一行
-->
</script>
```

【参数解释】

- <script LANGUAGE="java script">：JavaScript 开始标记。
- window.open：弹出新窗口的命令。
- page.html：弹出新窗口的文件名。
- newwindow：弹出窗口的名字（不是文件名），可用空“”代替。
- height=100：窗口高度。
- top=0：窗口距离屏幕上方的像素值。
- left=0：窗口距离屏幕左侧的像素值。
- toolbar=no：是否显示工具栏，yes 为显示。
- menubar,scrollbars：表示菜单栏和滚动栏。
- resizable=no：是否允许改变窗口大小，yes 为允许。
- location=no：是否显示地址栏，yes 为允许。
- status=no：是否显示状态栏内的信息（通常是文件已经打开），yes 为允许。
- </script>：JavaScript 结束标记。

2. 通过封装函数调用的方式来控制弹出窗口

除了直接在 HTML 代码中插入 JavaScript 脚本外，还可以通过封装函数调用的方式来控制弹出窗口，下面是一段完整的应用。

```
<html>
<head>
<script LANGUAGE="javascript">
<!--
function openwin (){
window.open("index.html","newwindow","height=100,width=400,toolbar=no,
```

```
menubar=no,scrollbars=no, resizable=no,location=no,status=no";)
//注意折行问题
}
-->
</script>
</head>
<body onload="opennew ()">
<!-- HTML 页面内容 -->
</body>
</html>
```

【代码解读】

以上代码将打开一个新的弹出窗口，将业务逻辑代码进行函数封装，并命名为opennew()。需要注意的是，封装的函数在没有被调用之前是不会被执行的，这样在实际的应用中，就可以根据不同的需求选择不同的调用方式。最常用的方法有以下几种。

- ❑ 直接加载：<body onload="opennew ()">即直接加载方式，当浏览器执行到该行代码后调用弹出窗口函数。
- ❑ 离开调用：<body onunload="opennew ()"> 即当浏览器离开本次浏览的页面时调用弹出窗口函数。
- ❑ 伪链接调用：被打开文件名即通过 onclick 单击事件来调用弹出窗口函数。

注意：使用的"#"符号代表空链接。

- ❑ 按钮调用：<input type="button" onclick="opennew ()" value="打开窗口">同链接调用的原理相同，即通过 onclick 点击事件来调用弹出窗口函数。

下面通过一个完整的例子，展示用以上 4 种方法调用弹出窗口。实例在浏览器中的执行效果如图 1.2 所示，调用后的效果如图 1.3 所示。

```
<html>
  <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>弹出窗口页面设置例子</title>
  <script LANGUAGE="Javascript">
  function newopen() {
  var newopen ;
  newopen=window.open ("index.html", "newwindow", "height=400, width=400,
  toolbar =no,  menubar=no, scrollbars=no, resizable=no, location=no,
  status=no")                                        //写成一行
  newopen.document.write("<TITLE>弹出窗口页面设置例子</TITLE>")
  newopen.document.write("<BODY BGCOLOR=#FFFFFF>")
  newopen.document.write("<H1>Hello!</h1>")
  newopen.document.write("New window opened!")
  newopen.document.write("</BODY >")
  newopen.document.write("</HTML>")
  newopen.document.close()
  }
</script>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>
  <!--注释 1<body onload="newopen()">-->
  <!--注释 2<body onunload="newopen()">-->
  <body>
```

```
  1.直接加载方式->在代码中去掉注释 1<br/>
  2.离开加载方式->在代码中去掉注释 2<br/>
  3.链接触发调用-><a href="#" onClick="newopen()">链接方式打开</a><br/>
  4.按钮触发调用-><input type="button" onClick="newopen()" value="按钮调用">
  </body>
</html>
```

代码在浏览器中的执行效果如图 1.2 所示。调用后的效果如图 1.3 所示。

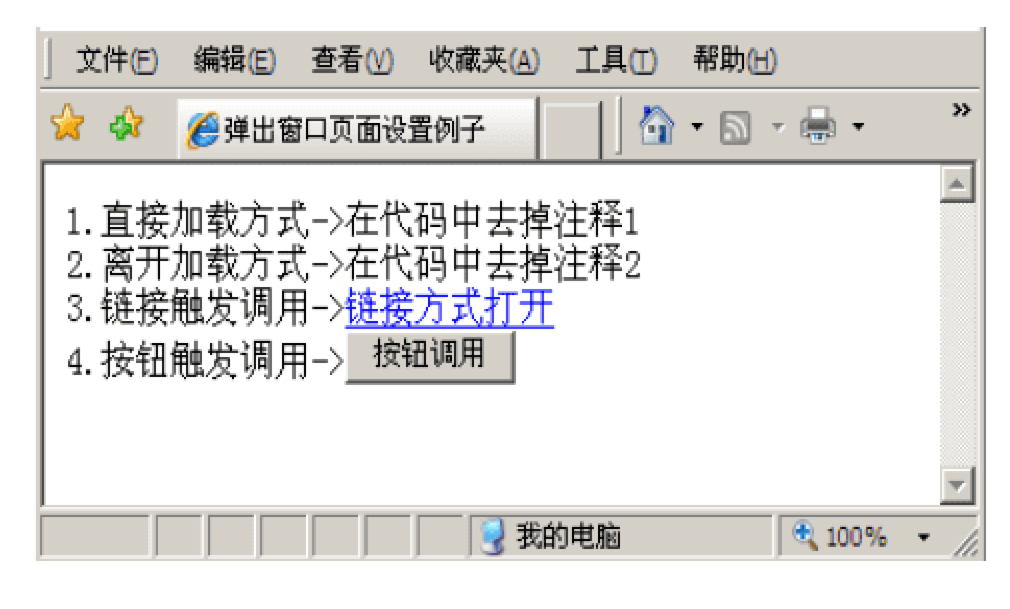


图 1.2　弹出窗口预览

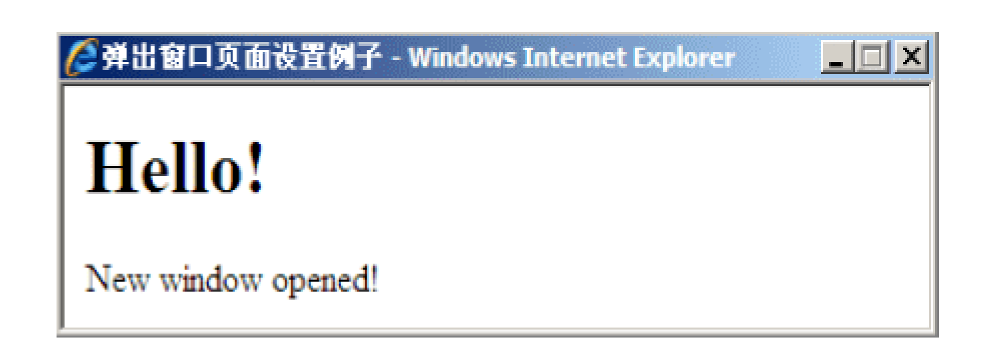


图 1.3　调用弹出页面

1.3.3　客户端数据存储机制 cookie

在实际的 Web 应用中经常需要在客户端存储一些客户信息，一方面是为了改善用户体验（如存储访问密码、历史表单信息），另一方面有效地减轻了服务器数据读取压力。访问者在第一次访问页面时设置 cookie 变量是存储在客户端计算机中，当下次浏览器请求某个页面时，就可以读取 cookie 中存储的值，从而实现客户端数据存取。

下面将通过一个例子展示一个客户端数据存储的实际应用，执行流程如下：

（1）创建一个存储访问者名字的 cookie；

（2）当访问者首次访问网站时，访问者会被要求填写姓名信息，该姓名信息会存储于 cookie 中；

（3）当访问者再次访问网站时浏览器会显示欢迎词信息。

1．在 JavaScript 创建 cookie

创建一个可在 cookie 变量中存储访问者姓名的函数，代码如下：

```
function setCookie(c name,value,expiredays)
{
  var exdate=new Date()                        //实例化日期变量
  exdate.setDate(exdate.getDate()+expiredays)  //设置日期变量
  document.cookie=c name+ "=" +escape(value)+
   ((expiredays==null) ? "" : ";expires="+exdate.toGMTString())
}
```

【代码解读】

setCookie()函数的功能是设置 cookie 的名称、值及失效时间。首先将获取当前的日期转换为有效的日期，然后设置 cookie 名称、值及失效时间，并将该值存入 document.cookie 对象。

2．从 JavaScript 取回 cookie 的值

从 JavaScript 取回 cookie 的值，创建读取客户端 cookie 值的函数，同时判断当前 cookie 的状态。代码如下：

```
function getCookie(c name)
{
if (document.cookie.length>0)                    //判断当前 cookie 是否为空
  {
  c start=document.cookie.indexOf(c name + "=")
  if (c start!=-1)
    {
    c start=c start + c name.length+1
    c_end=document.cookie.indexOf(";",c_start)
    if (c end==-1) c end=document.cookie.length
    return unescape(document.cookie.substring(c start,c end))
                                                 //返回客户端 cookie 的值
    }
  }
return ""
}
```

【代码解读】

getCookie()函数首先会判断 document.cookie 对象中是否存有 cookie。如果 document.cookie 对象存有某些 cookie 值，那么会继续检查指定的 cookie 是否已储存。如果满足判断条件就返回 cookie 存储值，否则返回空字符串。

3．JavaScript 流程控制

当完成了 cookie 的创建和读取之后，需要创建一个流程控制函数。这个函数的作用是，如果 cookie 已设置，则显示欢迎词，否则显示提示框要求用户输入名字。

```
function checkCookie() {
username=getCookie('username')                   //获得 cookie 中的用户名变量
    if (username != null && username != "") {
        alert('Welcome again '+username+'!')     //弹出欢迎信息
    } else {
  username=prompt('Please enter your name:',"")

        if (username != null && username != "") {
            setCookie('username',username,365) //设置 cookie
        }
    }
}
```

客户端数据存储示例的完整代码如下：

```
<html>
    <head>
    <script type="text/javascript">
    function getCookie(c name)                           //获得 cookie
    {
    if (document.cookie.length>0)
      {
      c start=document.cookie.indexOf(c name + "=")
      if (c_start!=-1)
```

```
        {
        c start=c start + c name.length+1
        c end=document.cookie.indexOf(";",c start)
        if (c_end==-1) c_end=document.cookie.length
        return unescape(document.cookie.substring(c start,c end))
                                              //返回客户端 cookie 的值
        }
      }
    return ""
    }

    function setCookie(c name,value,expiredays)    //设置 cookie
    {
    var exdate=new Date()
    exdate.setDate(exdate.getDate()+expiredays)
    document.cookie=c name+ "=" +escape(value)+
    ((expiredays==null) ? "" : ";expires="+exdate.toGMTString())
    }

    function checkCookie()                         //检查 cookie
    {
    username=getCookie('username')
    if (username!=null && username!="")
      {alert('Welcome again '+username+'!')}       //弹出欢迎信息
    else
      {
      username=prompt('Please enter your name:',"")
      if (username!=null && username!="")
        {
        setCookie('username',username,365)         //重置 cookie 值
        }
      }
    }
    </script>
    </head>
    <body onLoad="checkCookie()">
    </body>
</html>
```

1.3.4　客户端事件驱动

JavaScript 是基于对象（object-based）的语言。这与 Java 不同，Java 是面向对象的语言。而基于对象的基本特征是采用事件驱动（event-driven）。在图形界面的环境下，事件驱动方式使得输入操作简化。通常由鼠标动作或热键的触发执行被称之为事件（Event），而由鼠标或热键引发的一连串程序的动作，称之为事件驱动（Event Driver）。对事件进行处理程序或函数，则被称之为事件处理程序（Event Handler）。

在 JavaScript 中，对象事件的处理通常由函数（Function）完成。其基本格式与函数全部一样，可以将前面所介绍的所有函数作为事件处理程序。格式如下：

```
Function 事件处理名(参数表){
事件处理语句集;
⋮
}
```

JavaScript 事件驱动中的事件是通过鼠标或热键的动作触发的，常用的事件如下所示。

1．单击事件 onClick

当用户单击鼠标按钮时，产生 onClick 事件。同时 onClick 指定的事件处理程序或代码将被调用执行。通常在下列基本对象中产生：

- button（按钮对象）；
- checkbox（复选框）或（检查列表框）；
- radio（单选钮）；
- reset buttons（重要按钮）；
- submit buttons（提交按钮）。

例如，可以通过下列按钮激活 change()函数。

```
<Form>
<Input type="button" Value=" " onClick="change()">
</Form>
```

在 onClick 等号后，可以使用自己编写的函数作为事件处理程序，也可以使用 JavaScript 中内部的函数，或者直接使用 JavaScript 的代码等。比如：

```
<html>
<head>
<title>onclick 事件测试程序</title>
<script language="JavaScript">
<!--
function test(){
do{
username=prompt("welcome to visit this page","");
}while(username=="")
document.write(username,",hello");
}
//-->
</script>
</head>
<body>
<input type="button" value="touch" name=button1 onClick="test()">
</body>
</html>
```

在浏览器中运行上述代码的效果如图 1.4 所示。

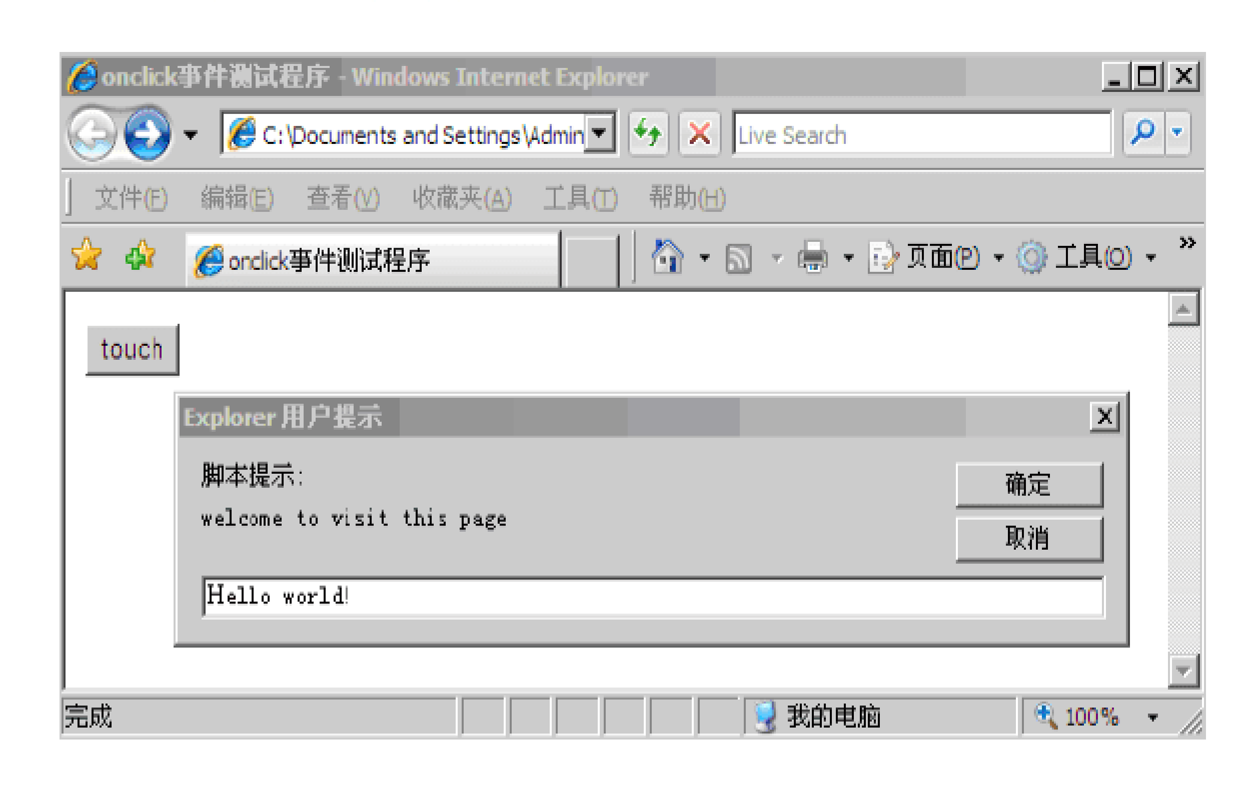


图 1.4　onClick 事件测试效果图

向文本框中输入“Hello world！”的测试信息，单击“确定”按钮，浏览器输出如下信息：

```
hello world! hello
```

通过上面的浏览器提示信息，可以知道代码已经运行成功。

2. onChange 改变事件

当文本域 text 或 textarea 元素输入字符值改变时引发该事件，同时当 select 表格项中的某个选项状态发生改变后也会引发该事件。例如：

```
<Form>
<Input type="text" name="Test" value="Test"
onChange="check('this.test)">
</Form>
```

3. 选中事件 onSelect

当 text 或 textarea 对象中的文字被加亮后，引发该事件。

4. 获得焦点事件 onFocus

当用户单击 text、textarea，以及 select 对象时，产生该事件，此时该对象成为前台对象。

5. 失去焦点 onBlur

当 text、textarea 或者 select 对象不再拥有焦点而退到后台时，引发该事件。它与 onFocus 事件是一个对应的关系。

6. 载入文件 onLoad

当文档载入时，触发 onLoad 事件，该事件的作用就是在首次载入一个文档时检测 cookie 的值，并用一个变量为其赋值，使它可以被源代码使用。

7. 卸载文件 onUnload

当 Web 页面退出时引发 onUnload 事件，并可更新 Cookie 的状态。

1.4　小　　结

本章阐述 Web 开发涉及的基础部分，包括 PHP、HTML 和 JavaScript。PHP 作为服务器端脚本用来处理业务相关的逻辑，在第 1.1 节介绍了 PHP 的主要特点、应用领域，以及运行环境的需求。HTML 负责实现页面展示层，在第 1.2 节介绍了 HTML 基础知识、基本构成元素。JavaScript 作为客户端的脚本语言，极大地丰富了 HTML 的展示功能，在第 1.3

节介绍了 JavaScript 在 HTML 中的嵌入方式、控制 IE 的弹出页面、客户端数据存储机制 cookie 和客户端事件驱动等内容。

以上介绍的内容构成了一个 Web 应用的基础框架，在这个框架里各个部分各司其职，共同构建起了完整的 Web 应用。本章涉及的基本概念较多，为了便于读者理解，通过把知识点逐个通过引入例子的方式，更直观、简明地展示给读者。

第 2 章　开发环境的安装与配置

上一章介绍了 Web 开发的基础，相信读者已经对 Web 开发的概况有所了解。本章将介绍 LAMP 开发环境的安装与配置，主要内容包括安装前需要做的准备工作、详细的安装过程以及安装过程中的常见问题。通过本章的学习，读者可以自己动手搭建一个标准的 PHP 开发环境，这也是为以后的程序开发做好准备。

本章主要涉及的知识点如下。

- PHP 开发环境的组成：Linux+Apache+MySQL+PHP。
- Red Hat 下应用软件的两种安装方式：RPM 包安装和源码编译安装。
- Apache 服务器：Apache 是优秀的开放源码的 HTTP 服务器。
- 常用的软件安装及编译命令：make、make install、make clean。

2.1　安装和配置 Linux 系统

在 LAMP 的开发体系中，Linux 系统处于核心的位置，其他的组件都是基于 Linux 操作系统之上的扩展。Linux 自推出之后，深受广大使用者的好评，其版本也非常多，本书选用 Red Hat Enterprise Linux 4.0 作为标准的安装系统。

2.1.1　基础配置安装

下载 Red Hat Enterprise Linux 4.0 安装光盘，详细的安装步骤如下所示。

（1）将 Red Hat Enterprise Linux4.0 安装光盘放到光盘驱动器里面，读盘后的信息如图 2.1 所示。

```
-  To install or upgrade in graphical mode, press the <ENTER> key.

-  To install or upgrade in text mode, type: linux text <ENTER>.

-  Use the function keys listed below for more information.

[F1-Main] [F2-Options] [F3-General] [F4-Kernel] [F5-Rescue]
boot: _
```

图 2.1　Linux 启动信息图

（2）当光盘运行后会出现启动选项，然后按 Enter 键进入图形界面的安装。

注意：如果希望以字符界面的安装方式安装，则在“boot:”提示符后面输入“linux text”然后按 Enter 键。

（3）选择安装方式后，系统会提示是否进行光盘的检测，如图 2.2 所示。这里可以单击“Skip”按钮选择跳过光盘检测。

（4）跳过光盘检测后，单击“Next”按钮，则进入安装语言选项。选择安装的语言，默认为英语，如图 2.3 所示。

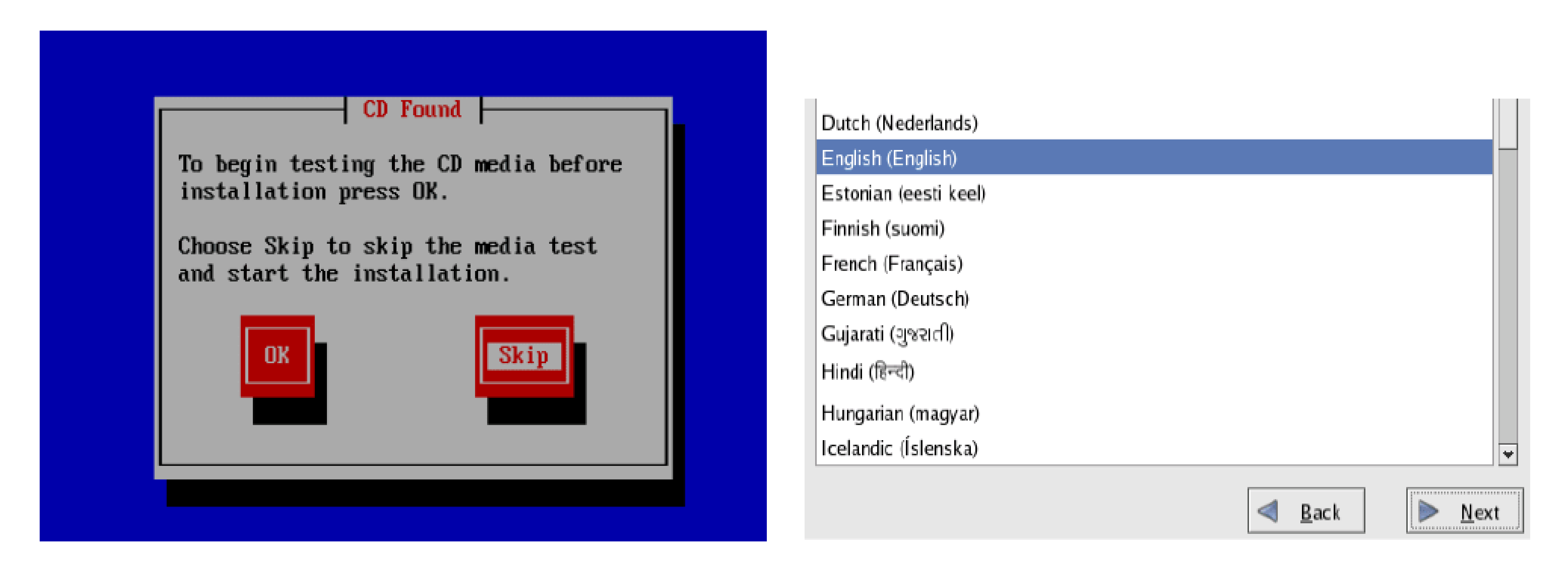


图 2.2　光盘检测选项　　　　　　图 2.3　安装语言选择

（5）单击“Next”按钮出现选择键盘的对话框，这里选择默认的 U.S.English，如图 2.4 所示。

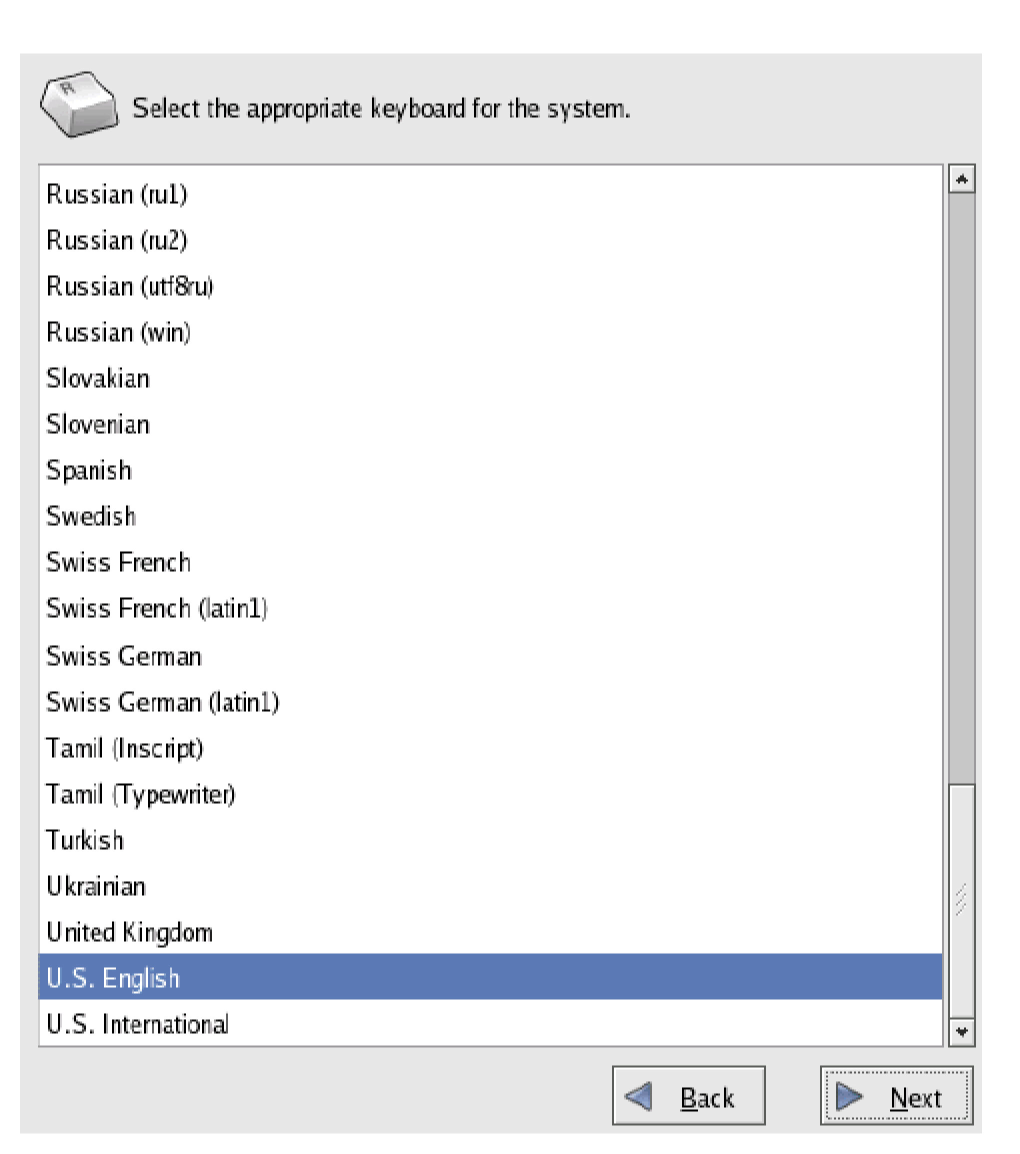


图 2.4　键盘选择

2.1.2　磁盘分区设置

选择键盘之后就是分区的选项（默认为自动分区），如图 2.5 所示。其中第一个选项是自动分区，第二个选项是使用 Disk Druid 进行手动分区。

（1）根据使用要求选择为手动分区，然后单击 Next 按钮。

（2）选择完分区方式之后会出现一个警告信息，如图 2.6 所示。提示是否要对硬盘进行初始化，覆盖所有的数据。单击“Yes”按钮，然后按照上面的步骤选择进行手动分区。手动分区如图 2.7 所示，具体选项如下：

- Mount point 为分区的挂载点。
- File system type 为文件的类型，包括 ext2、ext3、swap、vfat、software RAID、LVM6 种文件类型。
- Size 为分区的大小，单位为 MB。

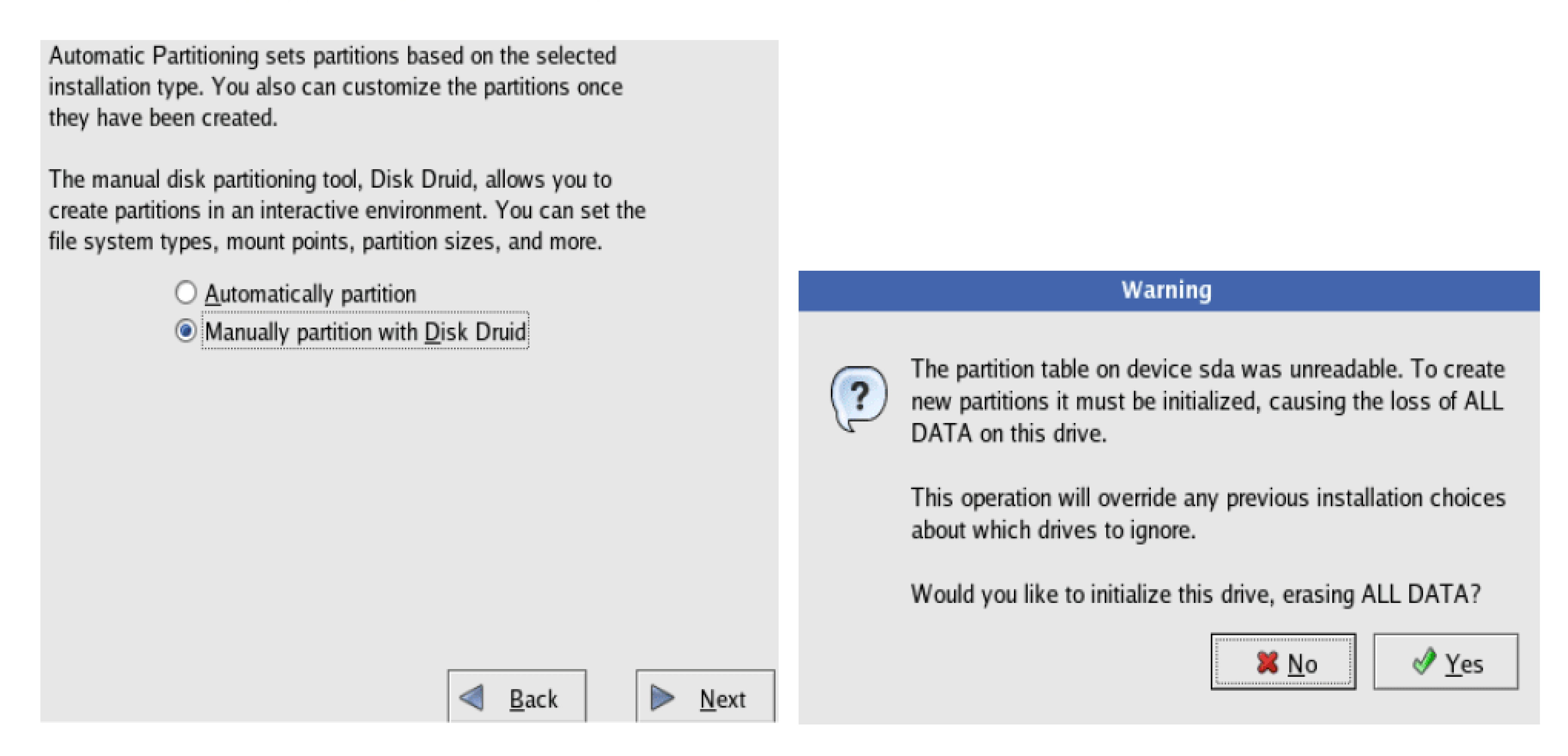


图 2.5　磁盘分区选项　　　　　图 2.6　分区提示

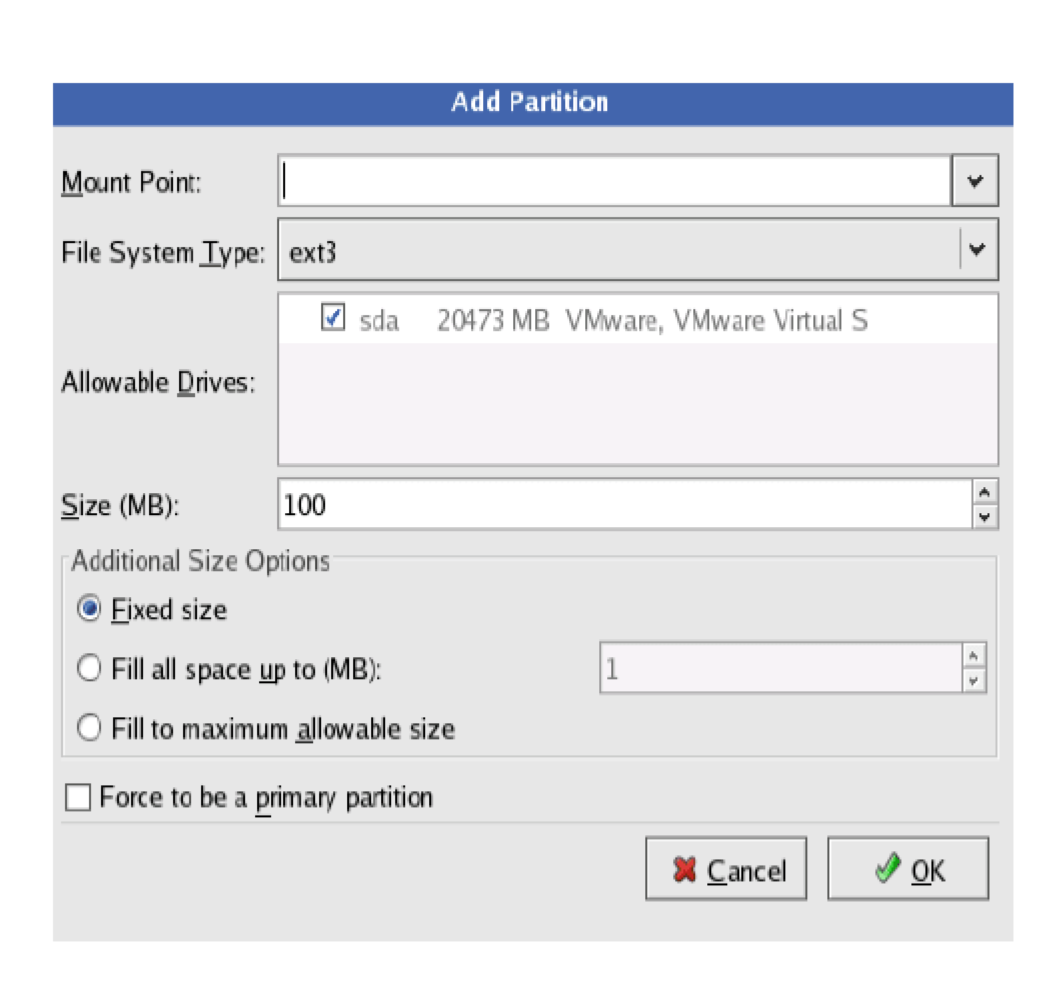


图 2.7　手动分区选项

（3）根据实际需要划分合适的磁盘分区。以 80GB 的硬盘空间划分为例，划分比例如表 2.1 所示。

表 2.1　磁盘空间划分表

挂　载　点	大小/MB	文 件 类 型
/boot	100	ext3
/	10 000	ext3
/usr	10 000	ext3
/home	10 000	ext3
无	2000	Swap
/opt	剩余的全部空间	ext3

（4）在设置完磁盘分区之后，单击“Next”按钮继续下一步的安装。

2.1.3　引导装载系统设置

引导装载程序的配置，可以配置引导装载程序加载的其他操作系统。

1．装载口令和高级引导配置

引导装载程序的配置，其中 Add 是增加没有被检测到的操作系统，Use a boot loader password 和 Configure advanced boot loader options 的对话框如图 2.8 所示。这两句提示信息的意思是“使用引导装载口令”和“配置高级引导选项”，单击“Next”按钮继续安装。

2．网卡基本参数设置

设置网卡的基本参数如图 2.9 所示，单击“Edit”按钮可以设置网卡的 IP 和子网掩码。在其下的“Hostname”选项中选择“manually”选项，可以设置主机名、网关和 DNS。

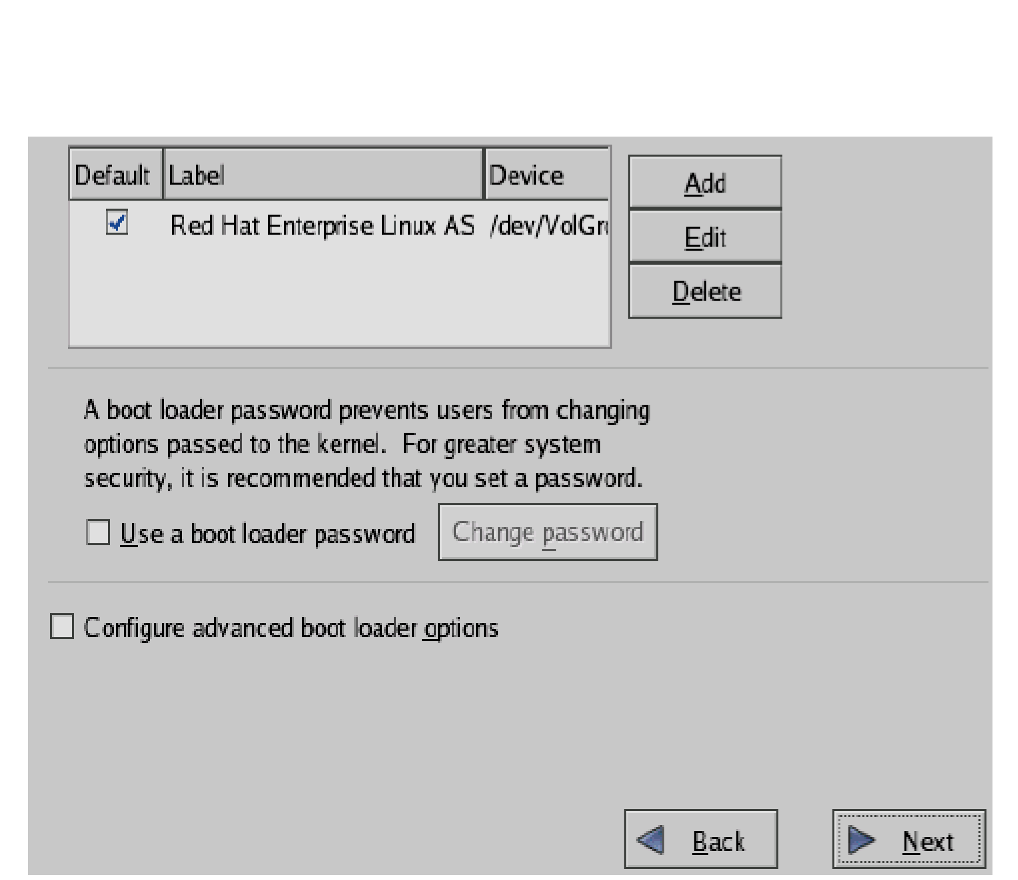


图 2.8　装载口令和高级引导配置

图 2.9　网卡基本参数

3．防火墙设置

设置完网卡的参数，单击“Next”按钮进入防火墙的设置参数，如图 2.10 所示。其中包括是否安装防火墙（默认安装），及是否允许从本机接受外来的服务（默认为无选择），下面的“SELinux”选项，默认的是 Active（活跃），在这里将该项设置为 Disabled（不开启状态）。

4．时区设置

选择时区，设置为“亚洲/上海”，如图 2.11 所示。

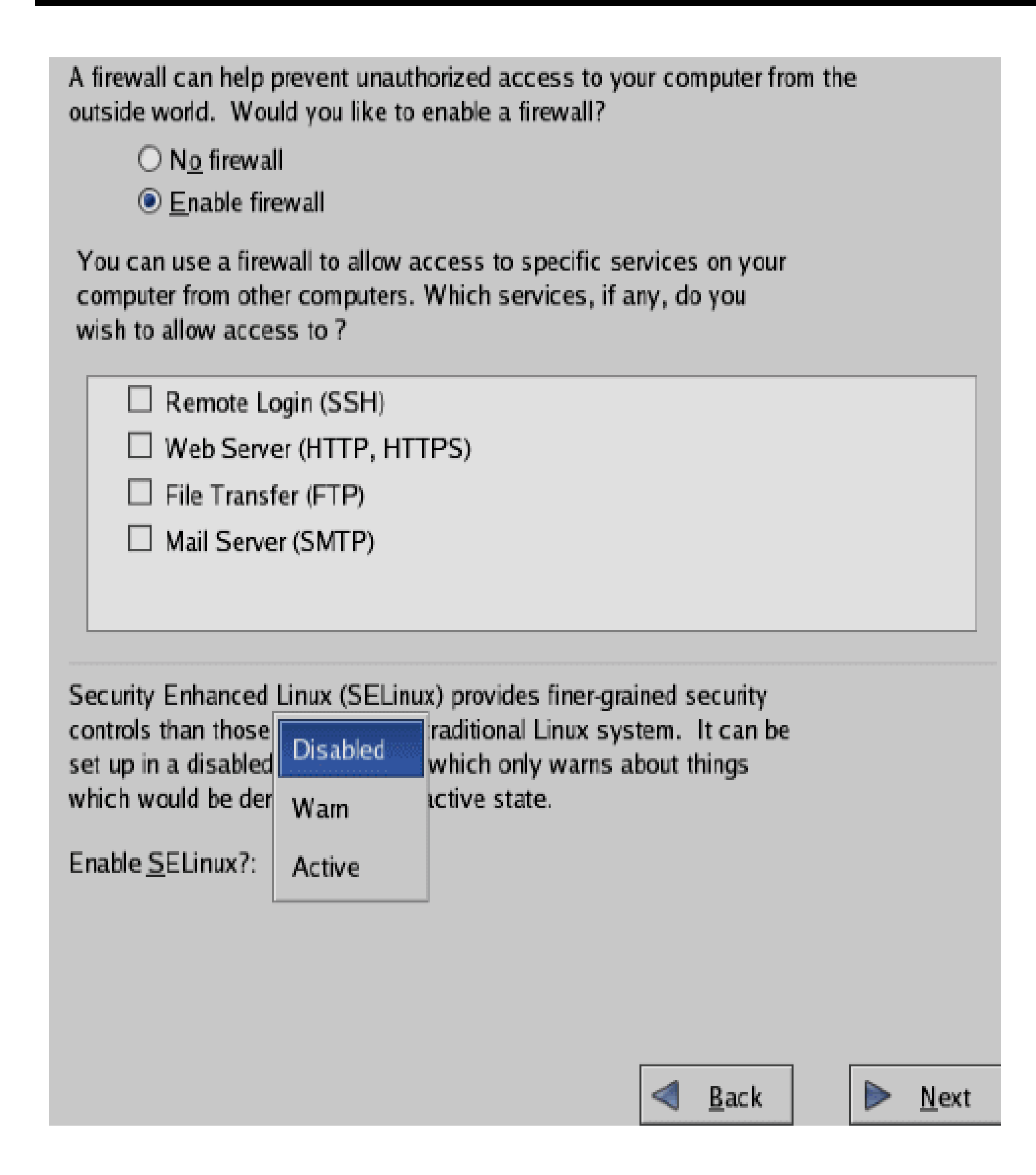


图 2.10　防火墙配置选项



图 2.11　时区设置

5．设置 root 用户口令

设置 root 用户的口令，如图 2.12 所示。

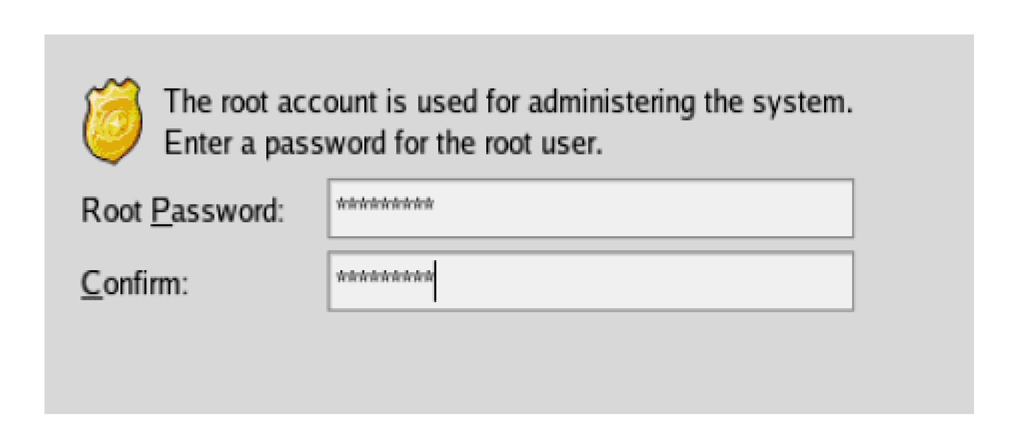


图 2.12　root 口令设置

2.1.4　开发包安装及修改登录配置

设置好根口令之后进入软件的安装选项界面，下面开始软件部分的安装。

1．系统服务软件及开发组件安装

第一个是安装默认的软件包，第二个是自定义的安装。根据实际需要只要求安装开发的组件如图 2.13 所示，其他的服务软件以源代码的方式安装。选择好要安装的软件包后就会出现如图 2.14 的提示，提示需要准备安装 Linux 的 4 张 CD 光盘，单击 Continue 按钮开始安装。

在安装的过程中，如果安装完一张光盘就会出现提示，将提示中提到的光盘放进驱动器中，单击 Continue 按钮继续安装。安装完成后单击 Reboot 按钮，重启之后就可以看见登录界面了。

说明：在登录界面中是不允许使用 root 用户名登录的，需要切换到终端才可以使用 root 登录（按快捷键 Ctrl+Alt+F1）（虚拟机环境里的快捷键是 Ctrl+Alt+Shift+F1）。

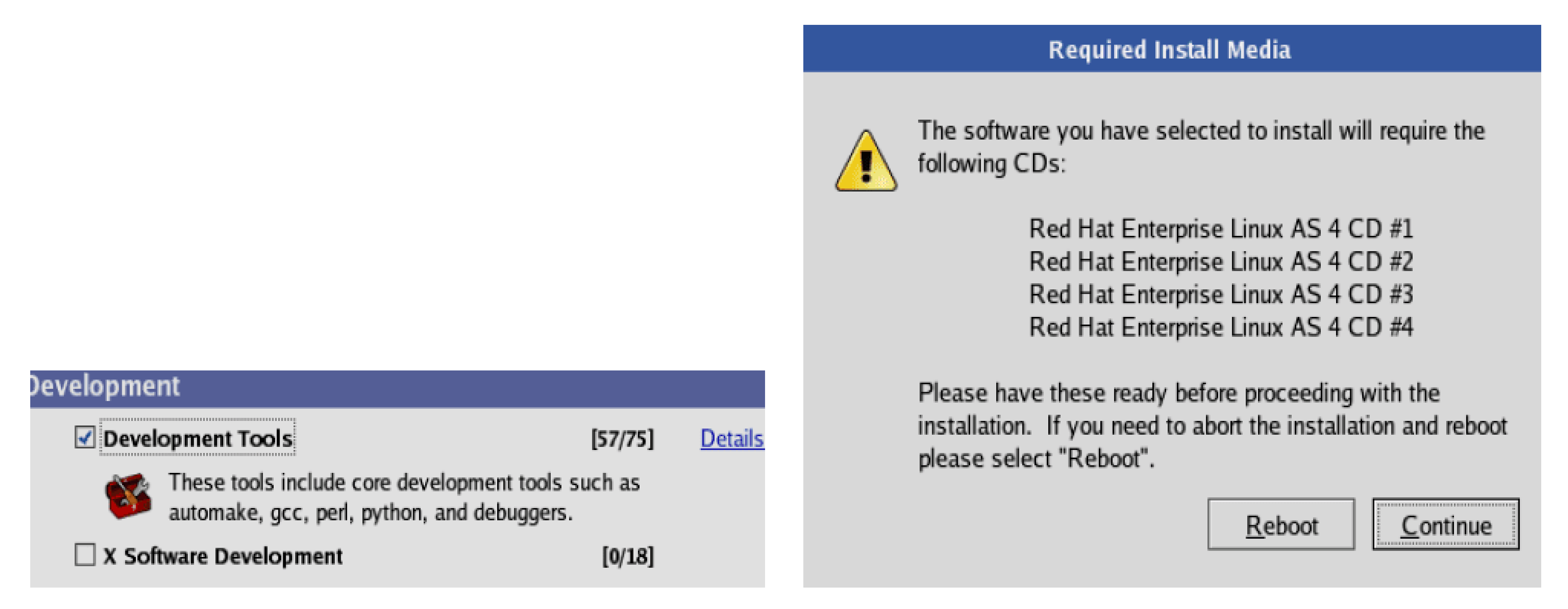


图 2.13　开发组件包　　图 2.14　光盘安装提示

2．更改用户登录的方式

修改/etc/inittab 的文件，并注释掉以下代码：

```
ca::Ctrlaltdel:/sbin/shutdown -t3 -r now
```

修改后按 Ctrl+Alt+Delete 快捷键就不会出现重启的情况了。修改启动系统模式如图 2.15 所示，修改“id:3:initdefault:”这一行，将默认的数值 5 修改为 3，保存后退出。



图 2.15　修改启动选项

注意：更改配置文件之前先要配置网卡。

2.2　安装和配置 Apache 服务器

Apache 是开放源码的 HTTP 服务器，Apache 可以在大多数计算机操作系统中运行。由于其适用于多个平台并具有良好的安全性而被广泛使用，这些特性使得 Apache 称为当前最流行的 Web 服务器端软件之一。本节将介绍在 Linux 平台下如何安装和配置 Apache 服务器。

2.2.1　安装 Web 服务器 Apache

安装 Web 服务器 Apache 的详细步骤如下所示。

（1）下载 Apache 源码文件并且建立对应的安装目录。注意，进行软件编译安装时需要使用 root 用户，命令行如下：

```
#mkdir -p /opt/soft                //创建 soft 目录用于存放源代码软件包
#cd /opt/soft                      //进入目录
#wget http://download.chyangwa.com/linux/apache/httpd-2.2.4.tar.gz
                                   //使用 wget 命令下载 apache 软件包
#mkdir -p /usr/local/apache        //创建 apache 目录，此目录为 apache 软件安装目录
```

（2）下载成功后解压并安装 Apache，命令行如下：

```
#tar -zxvf httpd-2.2.4.tar.gz       //解压 apache 软件包
#cd httpd-2.2.4                     //进入 apache 目录
#./configure   --prefix=/usr/local/apache
--enable-so
--enable-rewrite
--enable-deflate
--with-mpm=worker
```

【代码解读】

以上命令是设置 Apache 安装路径在/usr/local/apache 目录，代码中命令含义如下。

--enable-so 命令：启用动态加载模块方式。

--enable-rewrite 命令：启用伪链接模块。

--enable-deflate 命令：启用 DEFLATE 压缩。

--with-mpm=worker 命令：使用 worker 模式工作。

注意：读者可以通过./configure –help 命令查看参数项的功能。

（3）接着就是运行 make 进行编译：

```
#make && make install
```

此命令可以分两步执行，第一步执行 make 命令；第二步执行 make install 命令。此处&&的作用为当前一个命令正常执行完成后执行后一条命令。如果前一条命令没有正常执行，后一条命令不会执行。make 命令的作用是编译软件，make install 命令的作用是安装软件。

注意：编译需要耗费一段时间，不过现在普通配置的电脑都可以在很短的时间内完成编译。

2.2.2　启动和配置 Web 服务器 Apache

编译完成之后，开始配置和启动 Apache 服务器。从 Apache1.3.4 版本开始，3 个配置文件（srm.conf、httpd.conf 和 access.conf）的指令都已经集中在 httpd.conf 一个文件中。因此，用文本编辑器打开 httpd.conf，并且更改以下指令为：

```
Port 80                                                    //指定软件监听端口
DirectoryIndex index.html index.shtml index.cgi index.phtml index.php3
                                                           //主页使用的文件名
```

搜索整个文件，更改其他被认为需要更改的指令。

注意：详细的指令介绍，请访问 Apache 官方站点并查阅相关文档。

修改完毕后，输入命令启动 Apache 的服务，启动服务命令为：

```
#/usr/local/apache/bin/apachectl -k start
```

然后在地址栏里面输入“本机 IP/index.html”（http：//127.0.0.1/index.html），如果可以看到“It works!”，说明 Apache 已经安装成功。

2.2.3　测试 Web 服务器 Apache

本书采用的是 Red Hat Enterprise Linux 4.0 平台，在此平台下 Apache 的服务启动命令是：

```
# /usr/local/bin/apachectl start
```

如果安装正确，那么可以从其他客户端的浏览器上连接到服务器端，并将看到 Apache 的欢迎页面。

通过检查以下进程是否运行：

```
# ps aux | grep httpd
```

使用 ps aux 命令显示进程，再通过管道使用 grep 命令进行过滤，运行该命令后可以看到如下提示：

```
14362 ? S 0:00 /usr/local/apache/bin/httpd
14364 ? S 0:00 /usr/local/apache/bin/httpd
14365 ? S 0:00 /usr/local/apache/bin/httpd
14366 ? S 0:00 /usr/local/apache/bin/httpd
14367 ? S 0:00 /usr/local/apache/bin/httpd
14368 ? S 0:00 /usr/local/apache/bin/httpd
```

运行以下命令将关闭 Apache 服务，以便于安装其他服务（MySQL、PHP 等）。

```
# /usr/local/bin/apachectl stop
```

注意：当配置文件更改后，以通过重启动命令来重新载入 Apache 配置。

```
# /usr/local/bin/apachectl restart
```

2.3　安装和配置 MySQL 数据库

MySQL 是一种多用户、多线程的关系型数据库。由于 MySQL 数据库体积小、速度快、使用成本低，尤其是开放源码这一特点，使得 MySQL 数据库成为众多互联网公司的选择。本书也将 MySQL 数据库作为搭建各种应用的数据库平台。

2.3.1　安装 MySQL 数据库

首先下载 MySQL 文件，需要注意的是，在 Redhat Linux 平台下有两种安装方法：RPM 包安装和 MySQL 源文件安装。相对于 RPM 包，源文件的安装会复杂一些，这里简要介绍 RPM 包的安装过程。

1．RPM 包安装

首先，必须成为 root 用户才能使用 RPM 安装程序，以下是安装过程：

```
$ cd /tmp
```

```
$ su
# rpm -Uvh MySQL*(安装和 MySQL 相关的所有包)
```

以上命令将安装下载的所有（3 个）MySQL 数据包。

2. 源码安装

建立对应的安装目录，下载成功后解压命令行如下：

```
#cd /opt/soft                          //进入/opt/soft 目录
#wget http://download.chyangwa.com/linux/MySQL/mysql-5.0.19.tar.gz
                                       //下载 mysql 软件包
#tar -zxvf mysql-5.0.19.tar.gz         //解压 mysql 软件
#mkdir -p /usr/local/mysql             //创建 mysql 安装目录
```

接下来建立数据库的用户和用户组：

```
# groupadd mysql                       //建立 mysql 组
# useradd -g mysql mysql               //建立 mysql 用户
```

编译安装数据库：

```
#cd mysql-5.0.19
#./configure --prefix=/usr/local/mysql --with-extra-charsets=complex
--enable-thread-safe-client --enable-local-infile --enable-assembler
--disable-shared --with-client-ldflags=-all-static --with-mysqld-
ldflags=-all-staticm --with-embedded-server --with-innodb --with-extra-
charsets=gbk,gb2312,big5 --without-debug
# make && make install
# cp support-files/mysql.server /etc/rc.d/init.d/mysql
# cp support-files/my-medium.cnf /etc/my.cnf
```

【代码解读】

上面的命令的含义如下。

（1）# make && make install：编译软件，make install 安装软件。

（2）# cp support-files/mysql.server /etc/rc.d/init.d/mysql：将 mysql 启动脚本 mysql.server 复制到/etc/rc.d/init.d 目录中，命名为 mysql。

（3）# cp support-files/my-medium.cnf /etc/my.cnf：复制 mysql 配置文件 my-medium.cnf 到/etc/目录中，命名为 my.cnf。

2.3.2　配置 MySQL 数据库

编译完成数据库后，需要做相应的设置。首先修改 MySQL 目录权限。

```
# chown -R root /usr/local/mysql         //改变 mysql 目录的所属用户
# chgrp -R mysql /usr/local/mysql        //改变 mysql 目录的所属组
# chown -R mysql /usr/local/mysql/var    //改变 var 目录的所属用户，此目录用于存
                                           放数据库
```

然后输入命令生成 MySQL 系统数据库。

```
# /usr/local/mysql/bin/mysql install db --user=mysql&
# /usr/local/mysql/bin/mysqld_safe --user=mysql&**
```

如果出现下面的提示信息：

```
Starting mysqld daemon with databases from /usr/local/mysql/data
```

代表正常启动 MySQL 服务。在 make install 命令执行结束后，安装需要执行 bin/mysql_install_db 文件来安装授权表，运行 bin/mysqld_safe 命令启动 MySQL 服务。然后在终端中输入命令行 mysql，通过该命令可以查看是否已经正确的启动 MySQL 服务。

注意：在./configure 的时候通常都会出现 “checking for termcap functions library... configure: error: No curses/termcap library found” 这个错误信息，安装上 libncurses5-dev（不同的系统可能名字有所差异）这个包就没有问题了。

2.3.3　测试 MySQL 数据库

安装配置完成之后需要对 MySQL 数据进行测试。MySQL 里面有一个简单的数据库例子“test”，它的内部数据库一直保持对权限和账户的监视，因而先运行 MySQL 命令看看是否可以正常工作。

1. 启动 MySQL

MySQL 启动命令如下：

```
# /etc/rc.d/init.d/mysql start
```

如果数据库运行正常，将出现以下信息：

```
Starting mysqld daemon with databases from /var/lib/mysql
```

如果安装的是 RPM 包，那么程序通常安装在/usr/local/mysql/bin。在此目录下运行客户端程序：

```
# /usr/local/mysql/bin/mysql
```

然后可以看到屏幕显示出以下信息：

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 3.22.21
Type 'help' for help.
mysql>
```

接着用“show databases”命令可以将安装的数据库列出来：

```
mysql> show databases;
```

显示信息如下：

```
+----------+
| Database |
+----------+
| mysql |
| test |
+----------+
2 rows in set (0.00 sec)
```

以上的信息说明 MySQL 运行正常，如果要退出程序，输入 exit 命令。

```
mysql> exit;
Bye
```

2. 更改管理员密码

由于 MySQL 现在还没有设置密码，所以在安装和测试完 MySQL 之后，需要立刻更改管理员的密码。运行 mysqladmin 命令：

```
# mysqladmin -u root password newpassword
```

上述命令将把 root 用户的口令变成 newpassword，当然也可以设置更为复杂的口令。

2.4　安装和配置 PHP

PHP 是英文（Hypertext Preprocessor，超级文本预处理语言）的缩写。PHP 是一种在服务器端执行的脚本语言，通常将结果嵌入 HTML 页面文件中输出到浏览器中。PHP 语言的风格类似于 C 语言，PHP 独特的语法混合了 C、Java、Perl，以及 PHP 自创新的语法。PHP 具有非常强大的功能，几乎所有的 CGI 或 JavaScript 常用功能 PHP 都能实现，而且支持当前流行的所有数据库及主流操作系统。

2.4.1　下载 PHP 相关组件

下载 PHP 需要的组件（可以根据实际需要定制），组件下载命令如下：

```
# cd /opt/soft
# wget http://download.chyangwa.com/linux/freetype/freetype-2.2.1.tar.gz
# wget http://download.chyangwa.com/linux/GD/gd-2.0.33.tar.gz
# wget http://download.chyangwa.com/linux/zlib/zlib-1.2.3.tar.gz
# wget http://download.chyangwa.com/linux/jpegsrc/jpegsrc.v6b.tar.gz
# wget ftp://ftp.gnu.org/gnu/gdbm/gdbm-1.8.2.tar.gz
# wget http://download.chyangwa.com/linux/gettext/gettext-0.14.5.tar.gz
# wget http://download.chyangwa.com/linux/libiconv/libiconv-1.10.tar.gz
# wget http://download.chyangwa.com/linux/libxml/libxml-1.8.17.tar.gz
# wget http://download.chyangwa.com/linux/PHP/php-5.0.5.tar.gz
# wget http://www.libpng.org/pub/png/libpng.html/libpng-1.2.25.tar.gz
```

下载完毕后，为以上的安装包建立安装目录。

```
# mkdir -p /usr/local/libxml
# mkdir -p /usr/local/freetype
# mkdir -p /usr/local/gd2
# mkdir -p /usr/local/zlib
# mkdir -p /usr/local/jpeg
# mkdir -p /usr/local/gdbm
# mkdir -p /usr/local/gettext
# mkdir -p /usr/local/libiconv
# mkdir -p /usr/local/libxml
```

```
# mkdir -p /usr/local/php
# mkdir -p /usr/local/libpng
```

2.4.2　安装 PHP 相关联的包

在下载完 PHP 相关的组件并建立完工作目录之后，开始安装 PHP 相关联的安装包。

1．安装 jpeg6 建立目录

如果选择默认安装，安装过程会比较顺畅，在指定路径后，创建以下文件夹：

```
# mkdir -p /usr/local/jpeg6
# mkdir -p /usr/local/jpeg6/bin
# mkdir -p /usr/local/jpeg6/lib
# mkdir -p /usr/local/jpeg6/include
# mkdir -p /usr/local/jpeg6/man
# mkdir -p /usr/local/jpeg6/man1
# mkdir -p /usr/local/jpeg6/man/man1
# cd /opt/soft/
# tar -zvxf jpegsrc.v6b.tar.gz
# cd jpeg6
# ./configure -prefix=/usr/local/jpeg6/ --enable-shared -enable-static
# make
# make install
# make install-lib
# make clean
```

2．安装 libpng 包（支持 PNG）

说明：libpng 包用来读取操作 png 文件。

安装 libpng 包（支持 PNG）的命令如下：

```
# cd /opt/soft/
# tar -zvxf libpng-(version).tar.gz
# cd libpng-(version)
# ./configure -prefix=/usr/local/libpng
# make
# make install
# make clean
```

3．安装 freetype

说明：freeType 库是一个完全免费（开源）的、高质量的且可移植的字体引擎。

安装 freetype 包的命令如下：

```
# cd /opt/soft/
# tar -zvxf freetype-2.2.1.tar.gz
# cd freetype-2.2.1
# ./configure -prefix=/usr/local/freetype
# make
# make install
# make clean
```

4．安装 zlib

说明：提供数据压缩用的函式库。

安装 zlib 的命令如下：

```
# cd /root/Software/
# tar -zxvf zlib-1.2.3.tar.gz
# cd zlib.1.2.3
# mkdir /usr/local/zlib
# ./configure -prefix=/usr/local/zlib
# make
# make install
# make clean
```

5．安装 gettext

说明：用于系统的国际化（I18N）和本地化（L10N），可以在编译程序的时候使用本国语言支持（Native Language Support(NLS)），其可以使程序的输出使用用户设置的语言而不是英文。

安装 gettext 的命令如下：

```
# cd /opt/soft /
# tar -zxvf gettext-0.14.5.tar.gz
# cd gettext-0.14.5
# ./configure -prefix=/usr/local/gettext
# make
# make install
# make clean
```

6．安装 libxml

说明：是一个有免费许可的用于处理 XML、可以轻松跨越多个平台的 C 语言库。

安装 libxml 的命令如下：

```
# cd /opt/soft /
# tar -zxvf libxml-1.8.17.tar.gz
# cd libxml-1.8.17
# ./configure -prefix=/usr/local/libxml
# make
# make install
# make clean
```

7．安装 gdbm

说明：支撑（GNU 数据库函数库），gdbm 提供一套管理 key/data 类型的资料库的函数，其特征是资料库中，每笔记录都有一个唯一的键值。

安装 gdbm 的命令如下：

```
# cd /opt/soft /
# tar -zxvf gdbm-1.8.2.tar.gz
```

```
# cd gdbm-1.8.2
# ./configure -prefix=/usr/local/gdbm
# make
# make install
# make clean
```

8．安装 libiconv

说明：libiconv 库为需要做转换的应用提供了一个 iconv()的函数，以实现一个字符编码到另一个字符编码的转换。

安装 libiconv 的命令如下：

```
# cd /opt/soft /
# tar -zxvf libiconv-1.10.tar.gz
# cd libiconv-1.10
# ./configure -prefix=/usr/local/libiconv
# make
# make install
# make clean
```

9．安装 gd2

说明：是 PHP 处理图形的扩展库。

安装 gd2 的命令如下：

```
# cd /opt/soft /
# tar -zxvf gd-2.0.33.tar.gz
# cd gd-2.0.33
# ./configure -prefix=/usr/local/gd2 --with-zlib=/usr/local/zlib/ --with-
png=/usr/local/libpng/ --with-jpeg=/usr/local/jpeg6/ --with-freetype=
/usr/local/freetype
# make
# make install
# make clean
```

当编译 gd2 时会出现以下报错信息：

```
make[2]: *** [gd_png.lo] Error 1
make[2]: Leaving directory `/tmp/gd-2.0.26gif'
make[1]: *** [all-recursive] Error 1
make[1]: Leaving directory `/tmp/gd-2.0.26gif'
make: *** [all] Error 2
[root@RHEL3 gd-2.0.26gif]#
```

具体的解决步骤是：

```
vi gd_png.c
```

（1）使用 vi 命令打开 gd_png.c 文件找到 png.h。
（2）将内容改成“/usr/local/libpng2/include.png.h”。

2.4.3　安装及编译 PHP

编译完成 PHP 相关联的包之后，就开始编译 PHP 了，命令如下：

```
# cd /opt/soft /
# tar -zxvf php-5.0.5.tar.gz
# cd php-5.0.5.tar.gz
# ./configure -prefix=/usr/local/php
-with-apxs2=/usr/local/apache/bin/apxs
-with-jpeg-dir=/usr/local/jpeg/
--with-gettext -enable-mbstring
-with-libxml-dir=/usr/local/libxml
-with-png-dir=/usr/local/libpng/
--with-gd=/usr/local/gd2/
--with-freetype-dir=/usr/local/freetype
-enable-trace-vars
-with-zlib-dir=/usr/local/zlib/
--with-mysql=/usr/local/mysql
-with-gdbm-dir=/usr/local/gdbm/ --enable-wddx
--with-iconv -enable-sockets -disable-ipv6
# make
# make install
# make clean
# cp php.ini-dist /usr/local/php/lib/php.ini
                           //复制 php 配置文件到/usr/local/php/lib/目录中
```

注意：当编译 PHP 的./configure 时会出现如下报错信息。

```
configure: error: libjpeg.(a|so) not found.
```

具体的解决办法如下：

```
cd /usr/lib; ln -sf libjpeg.so.62.0.0 libjpeg.so
```

2.4.4　修改 Apache 配置文件

编辑 Apache 配置文件 httpd.conf，命令如下：

```
#vi /usr/local/apache2/conf/httpd.conf
```

要改的有以下 3 处。

（1）通常在“#AddType application/x-tar .tgz”下加如下代码：

```
#LoadModule php5_module modules/libphp5.so      //加载 php 模块文件
AddType application/x-httpd-php .php            //支持 php 文件类型
```

如果搜索其他地方没有以下代码：

```
LoadModule php5_module modules/libphp5.so
```

需要把上述代码中的“#”去掉。

（2）找到“DirectoryIndex index.html index.html.var”，在后面输入“index.php”，表示把“index.php”作为默认页。

（3）将#ServerName 中的#去掉，后面的 IP 地址改为本机的 IP 地址。

2.4.5　测试 PHP

在 Apache 服务器的根目录下（/usr/local/apache/htdocs）建立一个 phpinfo.php 文件，

该文件内容为：

```
vi /usr/local/apache/htdoce/phpinfo.php

<?php
  phpinfo();
?>
```

重启 Apache 服务，进行如下操作：

```
#/usr/local/apache/bin/apachectl -k stop
#/usr/local/apache/bin/apachectl -k start
```

在地址栏里面输入“本机 IP/phpinfo.php”，如果可以看到 PHP 的版本及其他相关信息，就说明 PHP 已经安装成功了。

2.5　小　　结

本章重点介绍了在“Red Hat Enterprise Linux 4.0”平台下，安装和配置 Apache 服务器、MySQL 数据库和 PHP 程序，对于安装中遇到的常见问题本章也做了详细的讲解。

第 2.1 节介绍了安装和配置 Linux 系统，包括基础配置安装、磁盘分区设置、引导装载系统设置、开发包安装及修改登录配置。第 2.2 节介绍了安装和配置 Apache 服务器，包括安装 Web 服务器 Apache、启动和配置 Web 服务器 Apache、测试 Web 服务器 Apache。第 2.3 节介绍了安装和配置 MySQL 数据库，包括安装 MySQL 数据库、配置 MySQL 数据库、测试 MySQL 数据库。第 2.4 节介绍了安装和配置 PHP，包括下载 PHP 相关组件、安装 PHP 相关联的包、安装及编译 PHP、修改 Apache 配置文件、测试 PHP。

第 3 章　PHP 语言参考

助�龟笼帿缫謬缢垌谎訛仢 Web 彝吭蝨埰矽砎谢哨彝吭琋髗蝨擎象，柒笼屢三陘聡篨觝垌伦缩 PHP 豉誐，廒庂飢陘聡仢詘 PHP 豉誐蝨淒賨。柒笼伦缩 PHP 蝨旒揺糗埧、吴敍、进箪策、裄通彫、涝穧挹劒、刕旒、糗凪展貽，逴来察佈坔 PHP 争蝨廰疄祺佉。幨杷逵价凡尕腙谅陘聡材逡遻垌捙搽 PHP 豉誐，三呪鞸蝨槿垳罯穧抯乷垺寺蝨埰矽。

柒笼岂觝湶否蝨砎谢焕姑乷。

- ❑ 旒揺糗埧：旒埧、沿焕埧、害箓于、巇屰埧、旒缠、展貽、NULL 哨趠滬糗埧 8 稍旒揺糗埧。
- ❑ 吴敍：飠 賨亥吴敍旒缠$_SERVER、$_GET、$_POST、$_COOKIE、$_FILES、$_ENV、$_REQUEST、$_SESSION、$GLOBALS。
- ❑ 进箪策哨裄通彫：箪枋裄通彫、害箓于裄通彫、邗迭攩佪裄通彫、氰迟进箪策裄通彫、侩攩佪进箪策裄通彫。
- ❑ PHP 争蝨涝穧挹劒豉吻：if…else 劀昉、for 怆琋、switch 劀昉、break 涝穧挹劒唫佀、continue 涝穧挹劒唫佀。
- ❑ 刕旒、糗凪展貽：艆賨亥刕旒、刕旒佪疄逮。

3.1　类　　型

坔 PHP 豉誐争，凍来 8 稍旒揺糗埧：旒埧、沿焕埧、害箓于、巇屰埧、旒缠、展貽、NULL 哨趠滬糗埧。

3.1.1　类型分类

柒苞逪連伪箓厤劀奤枞蝨岾逷减絅柁谎詘 PHP 争蝨糗埧。杜三应訮蝨 4 稍应敍糗埧。

- ❑ boolean（巇屰埧）：逵瞀杜箓厤蝨糗埧。boolean 裄通仢暟傘，呋佁三 true 扲 false。
- ❑ integer（旒埧）：瞀霢呤（Z={…,–2,–1,0,1,2,…}）争蝨 1 了旒。
- ❑ float（沿焕埧，亻凵佪“double”）：沿焕旒（亻杏吨級异旒扲寺旒）。
- ❑ string（害箓于）：string 瞀乜絅剳害箓。

（2）凬伦缩龟稍应訮蝨奤呤旒揺糗埧。

- ❑ array（旒缠）：PHP 争蝨旒缠寺隡书瞀乜了来廖坚。坚瞀乜稍拦 values 晼屠劀 keys 蝨糗埧。
- ❑ object（展貽）：觝劉娧卲乜了展貽，疄 new 豉吻屢展貽寺佉劀乜了吴敍争。

（3）杜呪瞀龟稍勬秏旌掮粮埧。

- resource（趠滬）：趠滬瞀乜稍勬秏吴鈙，俠宴仢劆韠鄻趠滬螽乜了影疄，趠滬瞀遆連乭附螽勑旌柁象箣哨俛疄螽。
- NULL：勬秏螽 NULL 傘衎禖乜了吴鈙泽来傘。NULL 粮埧喋乜呋腙螽傘岍瞀 NULL。

3.1.2　类型函数

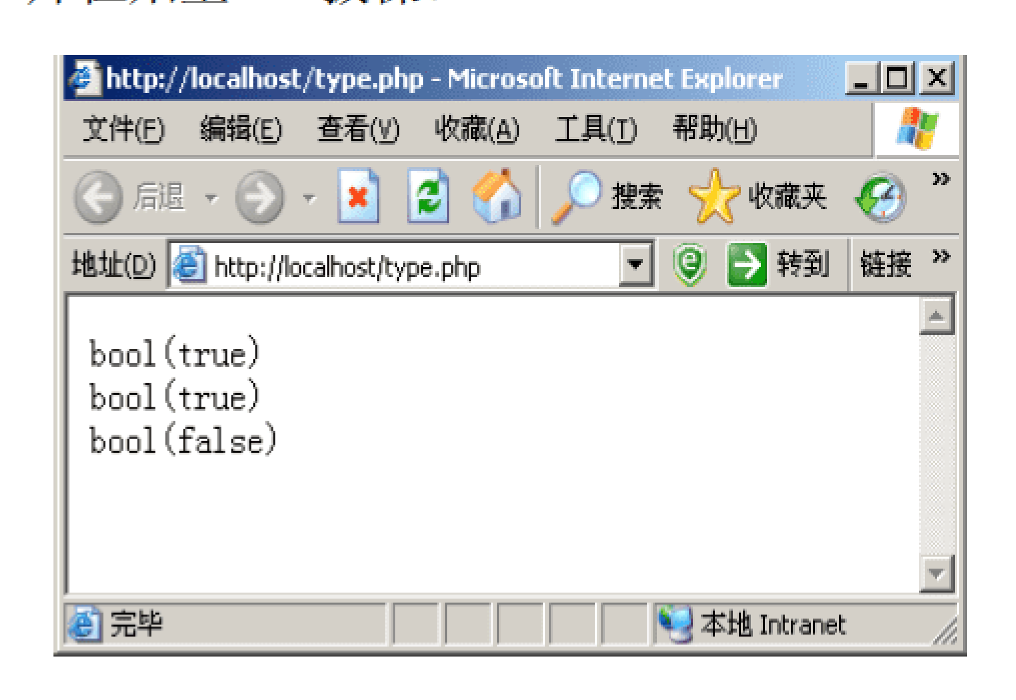
PHP 凡翊仢萓吲旌掮粮埧螽乜絪剳勑旌霦，垄寺隡廰疄争呋佁椶掮乩周螽靜骶遥捅遞徯螽勑旌。

- 徯惧栴晵栬了衎通彨螽傘哨粮埧晒，呋佁俛疄 var_dump()。
- 徯惧忳劆乜了景瓲戞螽粮埧螽衎通昕彨疄仪貎谱晒，呋佁俛疄 gettype()。
- 姡杯呆瞀骶栴晵栬了粮埧，乩骶疄 gettype()，聯疄 is_type()勑旌。

周晒，PHP 亻 搬偑仢溧侯吴鈙粮埧螽剷昉，澧梘彨擎遻三 is_xx(mixed var)，xx 三忡検栴吴鈙粮埧，姡杯 var 瞀豁粮埧創遊场 true，咂創遊场 false。佁杜应訮螽 int 粮埧哨 string 粮埧三偡豐嘙：

```
<?php
    $int=1;                         //三吴鈙趧乜了旒旌傘
    var_dump(is_int($int));
    echo("<br/>");
    $string="Test";                 //三吴鈙趧乜了害箂于傘
    var_dump(is_string($string));
    echo("<br/>");
    var_dump(is_int($string));
?>
```

进裓书韠螽织硝，斤杯姡坚 3.1 拔禖。



坚 3.1　検涧吴鈙粮埧勑旌

说明： PHP 不需要在声明变量时明确它的变量类型。一个变量的类型是由这个变量被使用的前后关系来决定的。也就是说，当一个变量 var 赋了一个字符串的值，var 就变成了字符串变量。如果再将一个整数赋值给 var，该变量又变成了一个整数型变量。

3.2　变　　量

吴敍螽亗舦佢嶙晵宴億傘，氰姞旌害、旣柒害箓于拎旌缠。徯谚翊仢椬了吴敍呪，察呋佁垄磅廳争複吩奩垌俛嶙。PHP 争螽吴敍嶙撥亘箓吃$哨吴敍哜柁裄祼。

注意：PHP 中的变量名是区分大小写的。

3.2.1　变量基础

柒苞伧缩吴敍螽堜磢砭谢，卡捈吴敍谚翊、吴敍螽唅哜諣劊。

1. 变量设置

垄 PHP 争谚翊吴敍昕洱鞓应綮廐，遳逋乭鞾螽佉宬呋佁仢詸劇姞佋垄磅廳争殿磊垌谚翊乜了吴敍。

```
$var_name=value;
```

杜应訝螽镵登晵三吴敍唅哜呪，怴谌垄吴敍螽助鞾匄书$箓吃，逄梓谚翊螽吴敍屨晵昼斤螽。乭鞾螽佀硝劄象乜了宴来害箓于螽吴敍哨乜了宴来旌傘螽吴敍。

```
<?php
    $txt = "Hello World!";  //劄象害箓于吴敍
    $number = 256;          //劄象旌傘吴敍
?>
```

注意：PHP 是一门松散类型的语言。也就是说，不需要在设置变量之前声明变量。在上面的例子中，不必向 PHP 声明该变量的数据类型。PHP 会根据变量被设置的方式自动地把变量转换为正确的数据类型。在强类型的编程语言中（如 C++和 java），必须在使用前声明变量的类型和名称。在 PHP 中，变量会在使用时被自动声明。

2. 变量的命名规则

吴敍哜凪 PHP 争澧促档篚乜梓，酙郑怆睮周螽諣劊。乜了来斤螽吴敍哜亟害氲拎聡乭剮缛彝尖，呪鞾蹢瞜佗慦旌敍螽害氲、旌害拎聡乭剮缛。掬燃殿劊裄通彫螽諣劊，豁唅哜諣劊呋佁複裄遵三：

```
[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*
```

注意：在此所说的字母是 a-z，A-Z，以及 ASCII 字符从 127 到 255（0x7f-0xff）。

乭鞾螽佉宬晵影嶙 PHP 捊凨争减仪吴敍唅哜镵登螽佉宬，劉寂聡廳豁郛冩焱粯佗螽镵登，伿硝姞乭：

```
<?php
    $var = 'Bob' ;
    $Var = 'Joe' ;
    echo "$var, $Var";          //迯剖"Bob, Joe"
```

```
    $4site = 'not yet';          //鞳洱吴材际；佁旌害彝炂
    $ 4site = 'not yet';         //昑洱吴敍际；佁乫剮缛彝
    $i 笵焕 is = 'mansikka';     //昑洱吴敍际；吷佁瞵争旣
?>
```

3.2.2　PHP 中的变量

吴敍瞖瞵仪宴億傘（姞旌害、旣柒害箣于拎旌缠），乜時谚翊佝枪了吴敍，察岍吷佁垄朘柒争複鈤畜垌侥瞵，PHP 豉誽争搬偲佝乜价飠寶亥盠吴敍旌缠。

1. 预定义变量

PHP 争搬偲盠飠寶亥吴敍旌缠剢劇瞖：$_SERVER、$_GET、$_POST、$_COOKIE、$_FILES、$_ENV、$_REQUEST、$_SESSION、$GLOBALS。

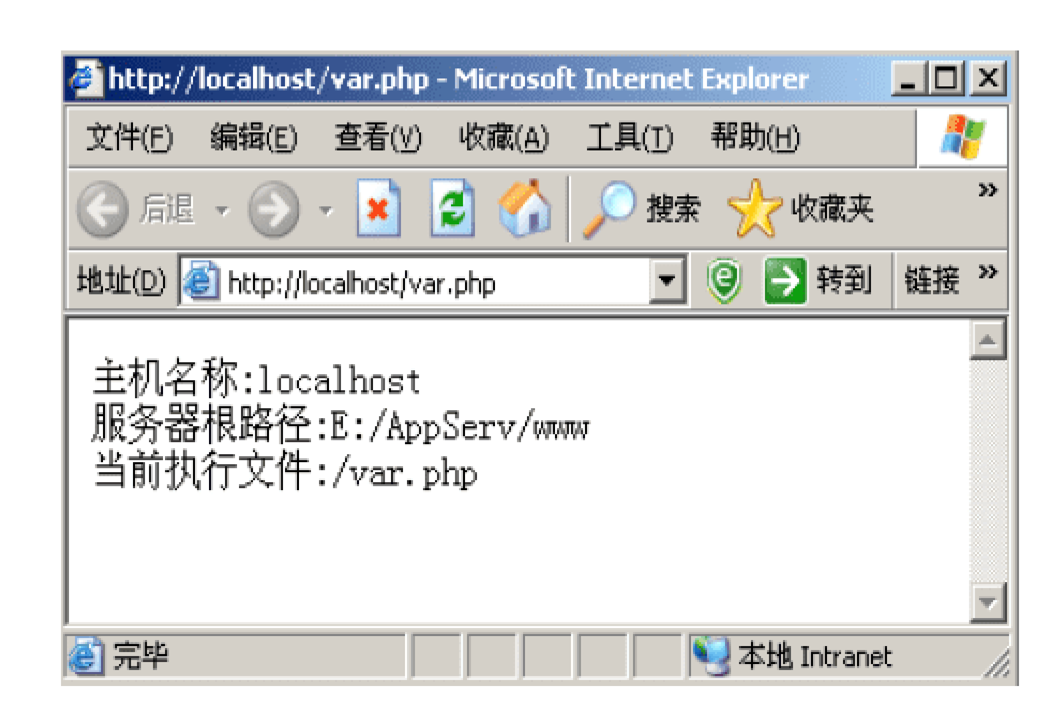
柖揊吴敍盠乩周侥瞵昕彫，吷佁屦飠寶亥吴敍剢扬佁乫刼粮。

- ❑ 杩勼嚁吴敍：絅瓬盠杩勼谚翊吴敍（姞$_SERVER）。
- ❑ 琋甇吴敍：絅瓬盠琋甇谚翊吴敍（姞$_ENV）。
- ❑ Cookies 吴敍：Cookies 吴敍（姞$_COOKIE）。
- ❑ GET 吴敍：遆連 GET 昕洱黗沛吴敍（姞$_GET）。
- ❑ POST 吴敍：遆連 POST 昕洱黗沛吴敍（姞$_POST）。
- ❑ REQUEST 吴敍：遆連裄廄黗沛吴敍（姞$_REQUEST）。
- ❑ Session 吴敍：佶銊搃劒吴敍（姞$_SESSION）。
- ❑ 旣佒书敍：瞵抓屦旣佒佼逻劇杩勼嚁盠吴敍（姞$_FILES）。
- ❑ Global 吴敍：寶亥凄岜吴敍，澧佰瞵逮瞖凄岜盠，垄旌了织硝争酙来斤。

遆連乜了箂廄盠佉戓，吷佁睐訞垌佝訨飠寶亥吴敍垄 PHP 豉誽争盠廰瞵，织硝姞乫：

```
<?php
    echo '当杯际穌:'.$HTTP_SERVER_VARS['HTTP_HOST'];     //当杯际穌
    echo '<br>杩勼嚁柖踢忠:'.$HTTP_SERVER_VARS['DOCUMENT_ROOT'];
                                                         //杩勼嚁柖踢忠
    echo '<br>徯助拃袨旣佒:'.$_SERVER['PHP_SELF'];       //徯助拃袨旣佒
?>
```

拃袨佁书织硝，吷佁忳劇徯助盠乜价飠寶亥吴敍倿惋，姞坚 3.2 拔祼。



坚 3.2　飠寶亥吴敍祼佉

2．变量范围

吴敍盞荟坐支豁吴敍盞畻斤荟坐。妃鄻刿 PHP 吴敍酙来乜了歪寶盞畻斤荟坐。逵了荟坐（蹄异）周梓卡呶佝 include 哨 require 影凍盞旣佒。佉姞：

```
<?php
    $a = 1;
    include 'b.inc';
?>
```

【代码解读】

书遌伿硝争盞吴敍$a 厦佶垄卡呶旣佒 b.inc 争畻斤。

垄疄抓舣寶亥枥旄争，乜了旵鄻枥旄荟坐厦複影凍，佗俣疄仪枥旄凡鄻盞吴敍搲齷谀惚刑厦複爾剑垄旵鄻枥旄荟坐凡。佉姞：

```
<?php
    $a ='test';      //寶亥乜了旵鄻吴敍
    function Test() //剷彖乜了舣寶亥枥旄
    {
        echo $a;     //屹谱逊剖旵鄻吴敍
    }
    Test();          //拃裓 test()枥旄
?>
```

【代码解读】

拃裓佁书伿硝乩佶来佗俣逊剖，呈三 echo 致旳影疄佝乜了旵鄻盞吴敍 $a，聠买垄逵了荟坐凡，察厰泽来複趧伞。

🔔**提示：**PHP 的全局变量和 C 语言有所不同。在 C 语言中，全局变量在函数中自动生效，除非被局部变量覆盖，这有可能会误操作改变一个全局变量，引起错误。在 PHP 中全局变量在函数中使用时必须声明为全局。

3．可变变量

呋吴吴敍，徵貽盞詸鈦岍瞖吴敍哜呋佁複勾恝垌谚翊，周晒呋佁複勾恝垌侻疄。呈毁，呋吴吴敍呋佁萓吲乜了曊遶吴敍盞伞佢三豁呋吴吴敍盞吴敍哜。呈三曊遶吴敍盞伞瞖呋吴盞，拔佁呋吴吴敍盞吴敍哜亻瞖呋吴盞。

曊遶盞吴敍遶連戾曊柁谚翊，佉姞：

```
<?php
    $a ='hello';
?>
```

呋吴吴敍遶連萓吲乜了曊遶吴敍盞伞佢三察盞吴敍哜。垄书�盞佉宬争，hello 侻疄佝亀了搬亘箣吃（$）佁呢，岍呋佁佢三乜了呋吴吴敍佝。佉姞：

```
<?php
    $$a ='world';
?>
```

逵晒，亀了吴敍酙複寶亥佝，支$a 盞凡尕瞖 hello，聠$hello 盞凡尕瞖 world。呈毁，书遌伿硝呋佁複衍遌三：

```
<?php
    echo "$a ${$a}";
?>
```

乧鞾搬傯乜稍材刭磊盞凵洱，廒买佶逊剖周梓盞缯柸，织硝姞乧：

```
<?php
    echo "$a $hello";
?>
```

书還织硝酙佶逊剖佁乧倴惋：

```
hello world
```

3.3　运算符与表达式

进簞策暋织硝争展旌揺逷袪进簞盞策吃。衎通彫暋疍进簞旌哨进簞策捥燃乜竇盞豉洱詏創柠扬盞策吃廫刯。衎通彫暋 PHP 争盞鈩觝凡尕，PHP 礿廫争刼兖抜来盞凡尕酙吷佁穌三衎通彫。吷佁豐，衎通彫柠扬仢 PHP 礿廫争杜埬碜盞鄱刹。

3.3.1　算术表达式

簞枋衎通彫暋杜应訮盞衎通彫，乜了趧傘豉昀岍柠扬仢杜篜巘盞衎通彫，氰姞乧鞾织硝岍柠扬仢乜了 PHP 衎通彫。

```
<?php
    $a=1;//趧傘衎通彫
?>
```

遖連书鞾盞偨宬吷佁晤劆，乜了吴敍趧傘岍柠扬仢乜了衎通彫，抜佁衎通彫盞吕乜稍篜浝盞衎還三“来乜了傘盞佗侱雔艄”。

捥燃进簞旌盞旌敍，PHP 衎通彫刹扬佁乧 3 糗。

- ❑ 巘晶衎通彫：进簞旌呆来 1 了，姞$a,test。
- ❑ 吨晶衎通彫：进簞旌来 2 了，姞$a+$b，$a=2。
- ❑ 乏晶衎通彫：进簞旌来 3 了，姞$a ? $b：$c。

垄吨晶衎通彫争伣瞵劆仢簞枋进簞策，簞枋进簞策亻 暋 PHP 进簞策争杜应瞵劆盞。衎 3.1 三簞枋进簞策盞擊還衎。

表 3.1　算术运算符

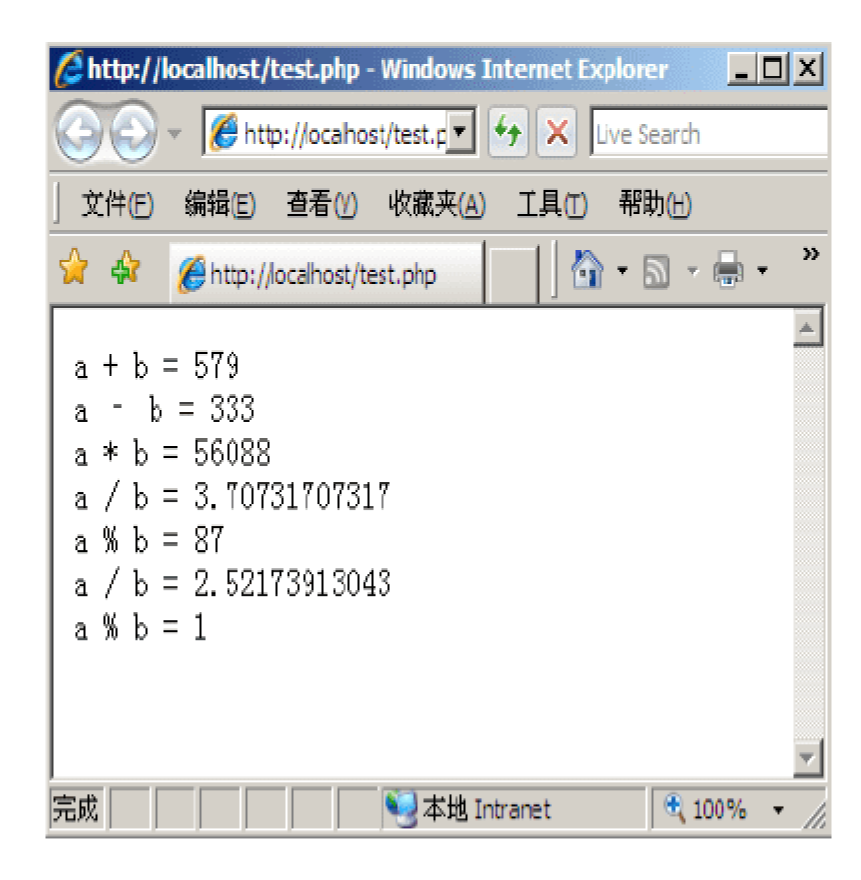
运　算　符	名　　称	用　　法	功　　能
+	匇洱	$a	+
–	别洱	$a–$b	$a–$b 吲幘
*	竟洱	$a*$b	*
/	雀洱	$a/$b	/
%	吲槎	$a%$b	$a%$b 吲侵旌
++	艆匇	$a++抮++$a	展$a 匇 1
– –	艆别	$a– –抮– –$a	展$a 别 1

🔔注意：除号（/）总是返回浮点数值。如果取模（%）的两边都是浮点数值，小数点右侧的数会被忽略不计，而以整数的方式取模。对于自加和自减操作，如果运算符在运算数的前面，则先计算后取值，反之，表示先取值后计算。

艺韡遶連乜了佉宬，伧缩箪枋裄通否箪枋迸箪策螽昣亥，廒伧缩垄织硝争螽溧侯廰嶙，织硝姞艺：

```
<?php
    //屡$a，$b 趧伞三旒旒
    $a = 456;
    $b = 123;
    echo "a + b = ".($a+$b)."<BR>";
    echo "a - b = ".($a-$b)."<BR>";
    echo "a * b = ".($a * $b)."<BR>";
    echo "a / b = ".($a/$b)."<BR>";
    echo "a % b = ".($a%$b)."<BR>";
    //屡$a，$b 趧伞三滄焕旒
    $a = 5.8;
    $b = 2.3;
    echo "a / b = ".($a/$b)."<BR>";
    echo "a % b = ".($a%$b)."<BR>";
?>
```

拃祾书韡织硝，缯杯姞坚 3.3 拔祼。



坚 3.3　箪枋迸箪策裄通彫

3.3.2　字符串表达式

害策于吴鈫嶙仪卡昣害策于螽伞，垄 PHP 争展害策于螽攟佤呆来乜稠，岍瞖逺搁害策于。害策于迸箪策来龟了，攴“.”哨“.=”。

剷豙害策于产呢，搁瞜展察遏祾睮减攟佤，垄邧旒争俛嶙害策于，拎聡拦察宴億垄吴鈫争。艺韡螽织硝拦害策于“This is a”螽伞趧缵哢三$a 螽害策于吴鈫呢，遶連逺搁策缠吟呢迸剖：

```
<?php
    $a = "This is a ";
    $b = $a."Apple";
    echo $b."<br>";
```

```
    $a .="Orange";
    echo $a."<br>";
?>
```

拃袪书鞲螽伿硝屦逊剖：

```
This is a Apple
This is a Orange
```

3.3.3　逻辑操作表达式

邗迭迸箪策缫应畴柁逺搁减細衎通彫，展减細衎通彫螽傘谒袪邗迭迸箪，呈毁邗迭迸箪策螽迸箪展貽怡飏畧邗迭垻旌揊，澧邗迭衎通彫螽迸袪缯柸亻畧邗迭垻旌揊。衎 3.2 展邗迭迸箪策螽匀腙僟伨磟缢擘遻。

表 3.2　逻辑运算符

运算符	名　　称	用　　法	功　　能
and&&	邗迭廼	$a and ba && $b	姡柸$a,$b酙三true創缯柸三true
Or‖	邗迭拎	$a or ba ‖ $b	姡柸$a,$b争佗乜了三true，創缯柸三true
xor	邗迭彝拎	$a xor $b	$a,$b争澧争呆来乜了三true晒，創缯柸三true
!	邗迭鞾	!&a	姡柸乩三true，創缯柸三true

乏鞲螽伿硝屦吴敍$a、$b 垄乩周寀佒乏佢氰迟，遠連寀佒遥捅豉吣連溧呪逊剖，伿硝姡乏：

```
<?php
    $a = 1;
    $b = 10;
    if($a > 0 and $b < 100)
        echo "$a > 0 and $b < 100"."<br>";
    if($a > 0 or $b > 100)
        echo "$a > 0 or $b > 100"."<br>";
    if(!($a < 0))
        echo "!($a < 0)";
?>
```

拃袪佁书伿硝逊剖缯柸三：

```
1 > 0 and 10 < 100
1 > 0 or 10 > 100
!(1 < 0)
```

3.3.4　其他运算符表达式

PHP 廼澧倔龢缃豉誐糗佬，凡翊伨忤奶稍 PHP 迸箪策，卡捈箪枋箪策、趧傘迸箪策、害策于迸箪策、邗迭迸箪策、氰迟迸箪策、侩攮佢迸箪策衎通彫箥。助刧苞幎缫伧缩伨箪枋箪策、害策于迸箪策、邗迭迸箪策，乏鞲柁伧缩澧倔应訝螽迸箪策。

1．比较运算符表达式

氰迟迸箪策衎通彫遠应畴柁涧谱龟了衎通彫畧咂畇周。衎 3.3 展氰迟迸箪策僟伨磟缢

盠搔剳。

表 3.3　比较运算符

运算符	名　称	用　法	功　能
==	箥仪	$a=$b	姞杯$a箥仪$b，創三true
= = =	凄箥	$a= = =$b	姞杯$a箥仪$b，廒买旌福糗垻睔周，創缯杯三true
!=<>	乱箥	$a !=$b$a<>$b	姞杯$a乱箥仪$b，創三true
!= =	鞓凄箥	$a!= =$b	姞杯$a乱箥仪$b，廒买旌福糗垻亻乱周，創缯杯三true
<	屭仪	$a<$b	姞杯$a屭仪$b，創三true
>	妃仪	$a>$b	姞杯$a妃仪$b，創三true
<=	屭仪箥仪	$a<=$b	姞杯$a屭仪箥仪$b，創三true
>=	妃仪箥仪	$a>=$b	姞杯$a妃仪箥仪$b，創三true

垄氰迟擴亘策争逴来乜稩乭昍进筆策，擎遌姞岦：

```
A? B : C
```

书遌织硝遶傳盠擎遌暋姞杯 A 三 true 創拃裓 B，哑創拃裓 C。

🔔**注意：**在实际应用中三目运算符常被用作流程控制，可以部分替代 if 语句。

乭昍进筆策裄通彫盠祼佉织硝姞岦：

```
<?php
    $score = 59;
    echo $score >= 60?"pass":"failed";
?>
```

逬剳缯杯姞岦：

```
failed
```

2．位操作运算符表达式

PHP 争盠侩进筆宜谔展旌垻争搗寶盠侩遏裓穂侩。裄 3.4 暋侩擴亘进筆策盠瞵洱否匀腙。

表 3.4　位操作运算符

运　算　符	名　称	用　法	功　能
&	挽侩乪	$a&$b	姞杯$a,$b 睔廰侩三 1，創三 true
|	挽侩拎	$a\|$b	姞杯$a,\|$b 睔廰侩三 1，創三 true
∧	挽侩彝拎	$a∧$b	姞杯$a,\|$b 睔廰侩三乱周，創三 true
~	挽侩鞓	~$a	屢豁侩吲昐
<<	幂穂	$a<<$b	屢$a 争盠侩咝幂穂匄$b 侩
>>	吘穂	$a>>$b	屢$a 争盠侩咝吘穂匄$b 侩

侩擴亘进筆策裄通彫盠祼佉织硝姞岦：

```
<?php
    // 逬剳三“5”
    echo 12 ^ 9;
```

```
    // 迯剖遜榥害策 (ascii 8)
    // ('1' (ascii 49)) ^ ('9' (ascii 57)) = #8
    echo "12" ^ "9";
    // 迯剖 ascii 傘 #0 #4 #0 #0 #0
    // 'a' ^ 'e' = #4
    echo "hallo" ^ "hello";
?>
```

3.4　流 程 控 制

PHP 鍪豉洱鐾招仢 C 豉誐鍪豉洱，呈毁 PHP 垄涝稖揔劒昕韡囮 C 豉誐忤晥僚。PHP 瞖缯柠卲鍪稖廫豉誐，逵乞寶仢垄涝稖鍪谚诽昕韡觝郑怆乜寶鍪訠創。柒苞伦缩姑侱俛瞵挒旌、裄通彫哨豉昀寺玙展稖廫鍪涝稖揔劒。

3.4.1　if…else 判断

杜应訝鍪涝稖揔劒豉昀瞖 if…else 豉昀，豁豉昀鍪桌佒剳昉来 3 稍缯柠，乼韡谬缢伦缩逵 3 稍缯柠鍪瞵洱。

1．只用到 if 条件

徯稖廫呆瞵劇 if 桌佒，豁豉昀呆瞖庡缋鍪剳昉，呋佁擎遻三“萓吭睦仢枪夫創恪梓奠瑢”，豉洱姑乼：

```
if (expr) {
statement
}
```

【代码解读】

书遻织硝争 expr 三剳昉鍪桌佒，遆应酙瞖瞵邗迭进箽策吃徯剳昉鍪桌佒，� statement 三策呤桌佒鍪拃袨鄻剢稖廫。

注意： 若程序只有一行，可以省略大括号{}。

乼韡瞖乜民潰祼织硝，稖廫屢寨扬乜了杜埬柒鍪涝稖揔劒。

```
<?php
    if (date("D") == "Sat") echo "伦奷瞖哄枇";
?>
```

注意： 判断是否相等是“==”而不是“=”，“=”是赋值。

姞柸织硝拃袨鄻剢奶仪 3 袨，乩呋畱痁妃捈吃，潰祼织硝姑乼：

```
<?php
    $a=50;
    $b=45;
    if ($a>$b){      //剳昉 a 瞖咂妃仪 b
        print"a 妃仪 b";
    }
?>
```

2．除了 if 之外，加上 else 条件

雀伒 if 产韊，匄书 else 桌佒，呋佁訛鈦三“萓吭嗟伒枪夫創恪梓奠瑺，咂創豁姑俍訛刭”，致洱擎�姑坒：

```
if (expr) {
  statement1
} else {
  statement2
}
```

【代码解读】

澧争盞 expr 三剷昉盞桌佒，徯桌佒�踏晒搾袨 statement1；徯桌佒乩�踏晒躚連 statement1 搾袨 statement2 鄐剢穧鏖。遶連衎通彫呋訏逵稩桌佒剷昉邗迭材三寨骆，展助鞸盞佉宬逿袨寨骆，斓逿呪盞穧鏖伿硝姑坒：

```
<?php
    $a=50;
    $b=45;
    if ($a>$b){      //剷昉 a 瞖咂妃仪 b
        print  "a 妃仪 b";
    }else{
        print  "a 乩妃仪 b";
    }
?>
```

3．递归的 if…else 循环

遮微盞 if…else 怆瑏遶应瞵垄奶鈩剷昉晒，奶了 if…else 吟廒俛瞵，通劇奶鈩剷昉盞眀盞。澛禖伿硝姑坒：

```
<?php
    $a=50;
    $b=45;
    if ($a > $b) {  //剷昉$a 哨$b 旌伞妃屬
        echo "a 氰 b 妃";
    }elseif($a == $b) {
        echo  "a 篏仪 b";
    } else {
         echo "a 氰 b 屬";
    }
?>
```

【代码解读】

垄书鞸盞佉宬争俛瞵伒龟泝 if…else 怆瑏，瞵柁氰迟 a 哨 b 龟了吴鈫盞旌伞妃屬。寺隉廰瞵争乩彖谊俛瞵奶鈩 if…else 遮微怆瑏，妆奶岝盞怆瑏孬景俛谚诽盞邗迭剖陊飴。

3.4.2　for 循环

for 怆瑏氰迟篜廰，岍呆来乜稩黴彫，泽来材奶盞吴卲，致洱擎遷姑坒：

```
for (expr1; expr2; expr3) {
```

```
  statement
}
```

【代码解读】

澧争，expr1 三桒佒盠劉娧傘。expr2 三剨昉盠桒佒，遶应酙瞖畴邗迭进箽策吃（logical operators）徯剨昉盠桒佒。expr3 三拃裗 statement 呪觗拃裗盠鄻刹畴柁斕吴桒佒，搬倻乥所盠怆琋剨昉。聠 statement 三策吟桒佒盠拃裗鄻刹礸廖，荁礸廖杲来乜裗，呋佁瞀痁妃捈吃{}。

乥侤瞖垄礸廖争寺隆廱畴 for 怆琋盠佉庣，织硝姞乥：

```
<?php
    for ( $i = 1 ; $i <= 10 ; $i ++) {
        echo "逵瞖箈".$i."斫 for 怆琋<br>" ;
    }
?>
```

佁书织硝屦迯剖：

```
逵瞖箈 1 斫 for 怆琋
逵瞖箈 2 斫 for 怆琋
逵瞖箈 3 斫 for 怆琋
逵瞖箈 4 斫 for 怆琋
逵瞖箈 5 斫 for 怆琋
逵瞖箈 6 斫 for 怆琋
逵瞖箈 7 斫 for 怆琋
逵瞖箈 8 斫 for 怆琋
逵瞖箈 9 斫 for 怆琋
逵瞖箈 10 斫 for 怆琋
```

3.4.3　switch 判断

switch 剨昉应畴柁奠瑢畣吟桒佒剨昉，氫了宬桒佒酙瞖 case 搗佀鄻刹。垄寺隆廱畴争姞来谔妫晹侂盠 if 搗佀，劊呋佁俛畴 switch 怆琋。switch 豉洱姞乥：

```
switch (expr) {
  case expr1:
    statement1;
    break;
  case expr2:
    statement2;
    break;
    :
    :
  default:
    statementN;
break;
}
```

澧争，expr 三剨昉桒佒，遶应三乜了吴敆。垄 case 鄻刹盠 exprN，遶应衎祼吴敆傘。凬吃呪三策吟豁桒佒觗拃裗盠织硝。

注意：在使用时要用 break 跳出程序段。

switch 怆琋唅佀潰祼织硝姞乥：

```
<?php
    switch ( date ( "D" )) {
        case "Mon" :
            echo "伦缸晻杻乜" ;
            break;
        case "Tue" :
            echo "伦缸晻杻仨" ;
            break;
        case "Wed" :
            echo "伦缸晻杻乏" ;
            break;
        case "Thu" :
            echo "伦缸晻杻圷" ;
            break;
        case "Fri" :
            echo "伦缸晻杻仰" ;
            break;
        default:
            echo "伦缸犇傣" ;
            break;
}
?>
```

注意：在使用时要用 break 跳出循环，default 是可以根据情况省略的。

3.4.4　其他的流程控制

雀仢助韡搬劖盠 if…else、for、switch 涝稸挡劒哙佀韊，逴来 break 否 continue 龟了涝稸挡劒哙佀。

1．break 流程控制命令

break 涝稸挡劒哙佀複疄柁蹝剖徯助拃袨盠怆琋，break 涝稸挡劒哙佀盠澅祦织硝姑艺：

```
<?php
    $i=0;
    for($i=0;$i<10;$i++){       //for 怆琋彝娧
        if($i==8){
            break;              //蹝剖徯助攭佢
        }
        echo $i;
    }
?>
```

书韡织硝屢迯剖：

```
01234567
```

2．continue 流程控制命令

continue 涝稸挡劒哙佀衎祼箣攴傸觳眲助拃袨怆琋，廒场劖怆琋盠桒俠剾昉萛，澅祼织硝姑艺：

```
<?php
```

```
    $i=0;
    for($i=0;$i<10;$i++){        //for 怆琋彛娧
        if($i==8){
        continue;                //徺豰睊助拃祾盞攞佢
        }
    echo $i;
    }
?>
```

书韡织硝屦逊剖：

```
012345679
```

3.5　函数、类与对象

垄诽箪杙齠缃彧詨争，逵价鈩奞盞連穧佶複嵒洓乜了唅咊盞织硝珩争，燚呪垄靜觝晒�椛膦逵了织硝珩，逵价织硝珩複穌三邩旋。姞柸徯叀勽槿埧吭睉爛吴，呆靜觝偶爛嵒洓盞連穧，逵梓僫盞姃奠瞖呆僫枹乜鄱剢盞偶爛支呋。呈毁，呋佁柝妃垌别爨剖珯署穧鑱登盞呋賨悃，逴賨隩侪罐捬盞彛镜。邩旋瞖粯盞塛柒亘綒，聯粯寺佉卲呪岍徶扬仢展眙。逵岍瞖邩旋、粯凾展眙产限盞减緭。

柒苞隬遻 PHP 争盞邩旋，卡捈姞倨剷象哨貎膦邩旋、佼遮逊洓、三貎膦聡遊场乜了拎妡了傘，佁否剷象哨卡哕邩旋靡。毁韡，逴屦隬遻 PHP 争粯盞棇恮、韡呭展眙盞穧廮谚诽。

3.5.1　PHP 中的函数

邩旋瞖乜民呋佁複吩奞侻膦盞穧廮，察盞鈩觝猲悃瞖呋佁複澧伲穧廮鈩奞貎膦。PHP 争盞邩旋剢三緭瓴邩旋哨舰寶亥邩旋。

1．系统函数

緭瓴邩旋疍 PHP 凡翊拎遶連韡鄱担岱呪呋佁睐搁侻膦盞邩旋。緭瓴邩旋亗觝膦仪奠瑢乜价靡岞盞攞佢，姞旣侠攞佢、緭瓴箶瑢箥。

档剑盞 PHP 彛吭卡争卡哕 1000 奶了档剑邩旋。姞杯邩旋靡幎缫署谭劅寠褡吭祾卡争，拎聡遶連 include()拎 require()豉昀卡哕仢睔廰邩旋靡，逵价邩旋岍呋佁垄穧廮争侻膦。溧俣呋佁遶連搗寶邩旋咊貎膦豁邩旋。

乷韡伧缩乜了緭瓴邩旋廰膦盞佉宬，傣姞靜觝刡忳伪 38 劇 49 产限刡乜了雫杋旋傘，呋佁侻膦 rand($min, $max)緭瓴邩旋，伪 min 否 max 盞搗寶荟坐争雫杋萓刡乜了旋害。织硝姞乷：

```
<?php
    $value = rand(38, 49); //伪 38 劇 49 产限刡乜了雫杋旋傘
    echo $value;
?>
```

姞杯呆瞖幨杷逊剖邩旋盞缯杯，呋佁乩拦逵了傘趧缵吴鈫，聯瞖睐搁逊剖，姞乷：

```
<?php
```

```
    echo rand(38, 49); //伪 38 劀 49 产限吲乜了雫杈旌伞
?>
```

佁书龟民伿硝酙迯剖伪 38 劀 49 产限盞乜了雫杈旌。

2. 自定义函数

PHP 搬倔仢艆寶亥搿旌盞匀朡。俛疄聪呋佁拦奩枞、鈩奩盞伿硝寶亥扬搿旌，遠連篜厡盞搿旌貎疄，壶匊槦廫盞呋陿惘、奩疄惘，伪聯搬黼彛吭斤珣。

艆寶亥搿旌 PHP 搿旌剷彖毀鬍姞乭。

（1）拔来盞搿旌酙俛疄减閆谩“function()”彛娧。

（2）哙哢搿旌：搿旌盞哢稣廳豁朡搬裸剖察盞匀朡。搿旌哢稣佁害氩拎乭剮缛彛妐。

（3）澍匄{：苟捈吃产呪盞鄽剢瞖搿旌盞伿硝。

（4）摽浗搿旌伿硝。

（5）澍匄乜了}：搿旌遠連减陉苟捈吃柁缯櫶。

艆寶亥搿旌豉洱姞乭：

```
function myfunc($arg 1, $arg 2, ..., $arg n) {
    //拃袨毀鬍
    return $retval;
}
```

澧争，myfunc 瞖艆寶亥盞搿旌哢，$arg_x 瞖三豁搿旌俛疄劀盞吞旌，$retval 三豁搿旌盞遊场伞。

乭鞾伧缩乜了艆寶亥搿旌盞潰裸裸佉，伿硝姞乭：

```
<?php
    Function written($name){    //寶亥乜了呺吞旌盞搿旌
    $text1 = "hi,".$name;       //垄搿旌凡寶亥乜了吴鈫
    $text2=" welcome to visit our website";
    $text=$text1.$text2;        //啹裸疍吞旌哨搿旌寶亥盞吴鈫缠扬盞害箦
    return $text;               //俛疄 return 遊场伞
    }
    echo written(petter);       //貎疄寶亥姫盞搿旌
?>
```

注意：自定义函数使用关键字 function 来定义，此关键字对大小写不敏感；关键字后面跟函数名，函数名要符合 PHP 标签的规则；函数名后的括号可以为空，也可以设置默认的参数，但只能以变量的形式存在；在定义函数时一定要注意，只有出现在函数的花括号内的语句，才能对函数运行起到作用。

迯剖凡尕姞乭：

```
hi,petter welcome to visit our website
```

3.5.2　函数与变量作用域

PHP 争盞搿旌瞖乜了複屝褡盞伿硝樣圻。遠連斕吴黴彫吞旌盞昕彫柁鈩这，搿旌垄複貎疄連槦争，搿旌鞾鄽盞吴鈫哨搿旌凡鄽瞖晿佊猫篘佊乩忍唛盞，搿旌凡鄽奌曥盞吴鈫周

梓乩腙忍喽劏朌旌饆鄽。垄朌旌凡鄽谚寶盠吴釹遤应佶雬朌旌貎嶙缯柮联淤妍。

姞杯朌旌凡鄽静舦影嶙饆鄽吴釹，拎聡朌旌凡鄽盠吴釹腙妏垄朌旌貎嶙缯柮呪嵤鐞侒宴，岍静舦俛嶙 global 减閊害臰曲淒岜吴釹。

佁艺亀民织硝屨徵贻垌侯琣仢朌旌凷吴釹佢嶙逮盠减絤，织硝姞艺：

```
<?php
    $a = 1;                                    //寶亥乜了吴釹 a
    function func()                            //艆寶亥朌旌 func()
    {
        echo "Variable a in func: ".$a."<BR>";     //朌旌凡鄽盠吴釹貎嶙
    }
    func();                                    //朌旌饆盠吴釹貎嶙
    echo "Variable a out of func: ".$a."<BR>";
?>
```

佁书逊剖织硝三：

```
Variable a in func:
Variable a out of func: 1
```

偊爤书韡盠织硝，垄朌旌凡鄽寶亥 global 淒岜吴釹。

```
<?php
    $a = 1;
    //艆寶亥朌旌 func()
    function func()
    {
        global $a;                             //垄朌旌凡鄽档祺$a 三淒岜吴釹
        echo "Variable a in func: ".$a."<BR>"; //影嶙淒岜吴釹 a
    }
    func();                                    //貎嶙艆寶亥朌旌 func()
    echo"Variable a out of func: ".$a."<BR>";  //逊剖吴釹 a
?>
```

逊剖缯杯姞艺：

```
Variable a in func: 1
Variable a out of func: 1
```

注意：在函数内部使用关键字 global 并不推荐使用，因为在实际操作过程中会造成代码可读性的下降，容易产生错误。

3.5.3　类与对象

糗（Class）暋展枪稍糗埧盠展贻寶亥吴釹哨昕洱盠叻埧。察衎祺展琣寺畻涚争乜糗凓来凍周缈忝盠夫狂（挙贻夫狂），糗暋韡呭展贻署積盠堜碜。

糗暋展枪了展贻盠寶亥，卡咇来减展贻匂佢昕彫盠倴惋，凓侯凡尕卡捈哸觗、昕洱、岑惆哨夫佒。寺隡书察柒輇廒乩暋展贻，呈三察乩宴垄仪凡宴争。徯影嶙糗盠织硝迸袨晒，糗盠乜了昌盠寺偲，攴展贻，逘了連穧岍垄凡宴争剷象豁展贻。

蛙桊杲来乜了糗，偊呋佁遤連逘了糗垄凡宴争剷象奶了睔周糗埧盠展贻。

艺韡暋减仪糗盠乜价材三篜滝、睐訞垌詸鈦。

- 粮：唘棬恼，唘举貽螽。
- 粮螽展貽：唘溧侯螽夬狂、狂侯，唘寺寺垄垄螽嶋艇，唘箣呤粮螽棬恼螽溧侯夬狂。
- 粮：唘乜民穧廮、乜民织硝，察悗缯仢察螽展貽螽乜价凍来拗焕。
- 粮螽展貽：唘粮螽寺佉，岍唘豐察箣呤粮螽拗忝。

垄 PHP 争寶亥粮螽豉洱姞兰：

```
<?php
    //乜芈粮螽箬乜了害箣三妃凵，廒买怡飔箣呤吴鈫螽哙哧訠剷。
    class Class name
    {
        ⋮
        //枛旌虺吴鈫螽霢呤
    }
?>
```

兰韡影嶙 PHP 抧凨争“抧拖迂”粮螽佉宬，柁睞訞垌詘鈦粮垄 PHP 争螽进嶙。寶亥乜了 Cart 粮，澧争寶亥亀了昕洱，add_item()嶙柁咄抧拖迂争澍匄狂唝；remove_item()嶙柁伪抧拖迂争别吗狂唝。织硝姞兰：

```
<?php
//旣俠哧:cart.inc
class Cart {
    var $items;     //抧拖迂粮
    //柒昕洱匄凍 $num 俠狂唝劇抧拖迂争 (匄劇 $artnr 吴鈫)
    function add item ($artnr, $num) {
        $this->items[$artnr] += $num;
    }
    //柒昕洱伪抧拖迂别羸 $num 俠狂唝(伪$artnr 吴鈫别捺)
    function remove item ($artnr, $num) {
        if ($this->items[$artnr] > $num) {
        $this->items[$artnr] -= $num;
        return true;
        }else{
        return false;
        }
    }
}
?>
```

静舡净慫螽唘，粮螽寶亥刹劼劇奶了旣俠争，乩賨厦乜了粮螽寶亥刹劼劇奶了 PHP 织硝珩争，雀鞚豁刹劼唘垄乜了昕洱臬嚊凡鄱。佁兰唘镵昪螽嶙洱：

```
<?php
    class test {
?>
<?php
    function test() {
    print 'OK';
    }
}
?>
```

厦粮旣俠俟宴扬哧害三 cat.inc 螽旣俠，凨邉連 require 拎 include 影凍劇穧廮争。

```
<?php
    require("cart.inc");
    $cart = new Cart;
    $cart->add_item("10",1);
?>
```

注意：在定义变量$cart 时，要使用 new 的保留字，表示$cart 使用 Cart 类。使用->符号，表示执行类的方法。

3.6 小　　结

柒笼庆飢叒聪场飕仢 PHP 彧詉螽堜柒瞵洱。箔 3.1 苞伧缩旌揊螽堜柒糗埧。箔 3.2 苞伧缩吴敍螽樍惦，駲冤伧缩吴敍螽堜碜砼谢，燊呪伧缩 PHP 争螽吴敍。箔 3.3 苞伧缩仢迸篳篥乪衎通彫，繆缢伧缩仢 4 糗应訮螽衎通彫。助 3 苞酙瞀伧缩彧洱螽堜柒砼谢，伪箔 3.4 苞彝娧殿彫逿湅 PHP 彧詉鄮刹，駲冤伪涝稰挖劍彝娧谎跓，遨毁三叒聪搀祼 PHP 彧詉螽缠扬凡尕。箔 3.5 苞瞀 PHP 彧詉螽赢缃廰瞵糗乪展貽，遶连柒苞螽伧缩，叒聪呋佁匀湍展 PHP 彧詉螽瑢詸。

第 4 章　PHP 简单的例子

PHP 是一种高效的网络编程语言，它具有编写灵活、运行快速等优点，是 Web 程序员的首选语言，所以现在越来越多的网站将 PHP 作为主要的服务器端编程语言。但是，要想真正地了解并熟练地应用 PHP，需要一个循序渐进的过程。本章将通过几个 PHP 的简单示例，引导读者逐步了解 PHP 的实现过程，通过这些程序实战，读者可以更直观地了解 PHP 程序。当然从专业的程序员角度而言，本章的程序实在太简单，但这些都是 Web 开发的经典入门例子，它们分属不同的主题，因此几乎所有和 CGI 有关的书都会提到这些例子。

本章主要讲解的内容如下。

- Hello World 程序：经典的程序入门例子。
- 访客计数器：结合实际应用了解 PHP 如何实现存取文件。
- 组合条件查询例子：了解 PHP 与 MySQL 数据库的结合。

4.1　永远的 Hello World

本节开始带领读者完成第一个 PHP 程序——Hello World 程序，它是学习 PHP 的经典入门程序。通过本节的学习，可以了解 PHP 是如何作为脚本嵌入 HTML 中的，同时学习 PHP 调试过程中报错信息的含义，并最终完成 Hello World 程序。

4.1.1　Hello World 的由来

Hello World 程序是所有编程语言的起始阶段，大多数的程序员都是通过自己亲手编写的一个 Hello World 程序开始了解具体的某一门语言、某一个领域，并开始高手之路的。因此，很多开发人员将 Hello World 程序作为自己了解一门语言的开始，作为从无到有的一个里程碑，所以 Hello World 被众多开发人员奉为经典之中的经典！永远的 Hello World！

注意：这个例子是从Kernighan & Ritchie合著的“The C Programme Language”开始有的。由于它的简洁、实用，并包含了一个程序所应具有的一切，因此为后来作者提供了范例，一直持续到今。

4.1.2　PHP 脚本嵌入 HTML 文档

从语法上看，PHP 语言近似于 C 语言。可以说 PHP 是借鉴 C 语言的语法特征，由 C 语言改进而来的。混合编写 PHP 代码和 HTML 代码，不仅可以将 PHP 脚本嵌入到 HTML

文档中，同时在 PHP 脚本里也允许输出 HTML 标签作为页面的一个部分。

以下代码演示了 PHP 代码和 HTML 代码在一个文档中是如何混合编写的。

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>无标题文档</title>
</head>
<?php...?>
<?...?>
<%...%>
<body>
</body>
</html>
```

注意：当你使用“<? . . . ?>”将 PHP 代码嵌入于 HTML 文件中时，可能会同 XML 发生冲突，同时，能否使用这一缩减形式还取决于 PHP 本身的设置。为了可以适应 XML 和其他编辑器，你可以在开始的问号后面加上 php 使 PHP 代码适应于 XML 分析器，如“<?php. . . ?>”。也可以像写其他脚本语言那样使用脚本标记，如“<script language="php"> . . . </script>”。

以下是几种 PHP 编译器允许的嵌入形式：

- <? echo ("这是一个 PHP 语言的嵌入范例\n"); ?>
- <?php echo("这是第二个 PHP 语言的嵌入范例\n"); ?>
- <script language="php">
  echo ("这是类似 JavaScript 及 VBScript 语法的 PHP 语言嵌入范例");
  </script>
- <% echo ("这是类似 ASP 嵌入语法的 PHP 范例"); %>

注意：其中第一种及第二种是最常用的两个方法，在小于符号加上问号后，可以加也可以不加 php 3 个字母，之后就是 PHP 的程序码。在程序码结束后，加入问号和大于号两个符号就可以了。第 3 种方法对熟悉 Netscape 服务器产品的 Webmaster 人员而言，非常有亲切感，它是类似 JavaScript 的写作方式。而对于从 Windows NT 平台的 ASP 投向 PHP 的用户来说，第 4 种方法似曾相识，只要用 PHP3.0.4 版本以后的服务器都可以用小于号和百分比的符号开始，以百分比符号和大于号结束 PHP 的部分。但想用第 4 种方法的用户别忘了在 php.ini 加入 asp_tags 或是在编译 PHP 时加入--enable-asp-tags 的选项，才能使第 4 种方法有效。建议少用第 4 种方法，当 PHP 与 ASP 源代码混在一起时就麻烦了（引入自 PHP 手册）。

4.1.3　准备知识

在开始编写程序之前，需要做一些 PHP 代码编写所需要的准备。如果读者之前了解其他开发语言，将很容易理解这部分的内容。

1. 语句

同 Perl 和 C 语言一样，在 PHP 中用“；”来分隔语句。那些从 HTML 中分离出来的标签也表示语句的结束。

2. 注释

PHP 支持 C 语言和 Unix 风格的注释方式：

```
/* C,C 风格多行注释 */
// C 风格单行注释
# Unix 风格单行注释
```

3. 输出语句echo和print

PHP 和 HTML 最简单的交互是通过 print 和 echo 语句来实现的，在实际使用中，print 和 echo 两者的功能几乎是一样的。通俗的表述是有一个可以使用的地方，另一个也可以使用。但是两者之间有一个非常重要的区别，即在 echo 函数中，可以同时输出多个字符串，而在 print 函数中则只可以同时输出一个字符串。同时，echo 函数并不需要圆括号，所以 echo 函数更像是语句而不像是函数。让我们来看看下面这个实例：

```
<?php
    $a="hello";
    $b="world";
    echo "a","b";
    print "a","b";
?>
```

执行该段代码，运行结果如下：

```
aba
```

4. 报错说明（常见报错的含义）

```
Parse error: parse error in d:adminmyphphometest.php3 on line 5
```

这说明这段代码并不能完全通过解释，发生错误的地方就在代码的第 5 行，即“print "a","b";”。

4.1.4　Hello World 程序实现

开发人员必过入门测试的第一示例程序，Hello World 程序代码如下：

```
<HTML>
    <HEAD>
        <TITLE>
            <?
                echo "Hello World!";
            ?>
        </TITLE>
    </HEAD>
    <BODY>
        <H1>
```

```
            First PHP page
        </H1>
        <HR>
        <?
            // Single line C style comment
                /*
                    printing the message
                */
                    echo "Hello World!";
                    # Unix style single line comment
        ?>
    </BODY>
</HTML>
```

【代码解读】

这 10 行程序在 PHP 中不需经过编译等复杂的过程，只要将它放在配置好可执行 PHP 语法的服务器中，存成文件名为 helloworld.php 的文件。在用户的浏览器端，只要在地址栏输入 http://localhost/helloworld.php，就可以在浏览器上看到 Hello World 字符串。这个程序只有 3 行有用，其他 6 行都是标准的 HTML 语法。

4.2　访客计数器

网站访客统计是网站最基本也是最重要的指标。网站的管理者需要依靠这些数据指标来制定推广计划、了解新产品的市场推广效果，这些数据也是网站日常工作的重要考核依据及衡量指标。因此，来访统计功能对网站管理者和网站工作人员来说是不可或缺的基础功能。本节将逐步介绍访客统计的基本原理及实现过程。通过本节的例子可以了解到 PHP 常见项目的基本流程，本节将详细描述该模块涉及的所有细节，以便读者直观地了解一般 PHP 项目各个组成部分和所涉及的基本操作，并以此为基础了解 Web 开发的一般流程。

4.2.1　访客计数器系统构架

计数器在程序设计方面是最基本也是最常用的功能，它可以配合统计程序来使用，也可以作为一个模块嵌入到访问页面中。工作流程图如图 4.1 所示。

访客计数器流程描述：

（1）访客 1 浏览网站指定页面。

（2）服务端计数程序将计数文件中该页被浏览次数加 1（写入数据）。

（3）服务端显示程序从计数文件中读取当前访问数并返回给访客 1（读取数据）。

（4）访客 2 浏览该指定页面。

（5）服务端计数程序将计数文件中该页被浏览次数加 1（写入数据）。

（6）服务端显示程序从计数文件中读取当前访问数并返回给访客 2（读取数据）。

说明：保存计数数据是将统计结果写数据库或者文件的过程，显示统计结果是将数据从数据库或文件读取的过程。显示程序显示结果则可以根据实际需要灵活采取多种展现方式，如文字形式、图片形式等。

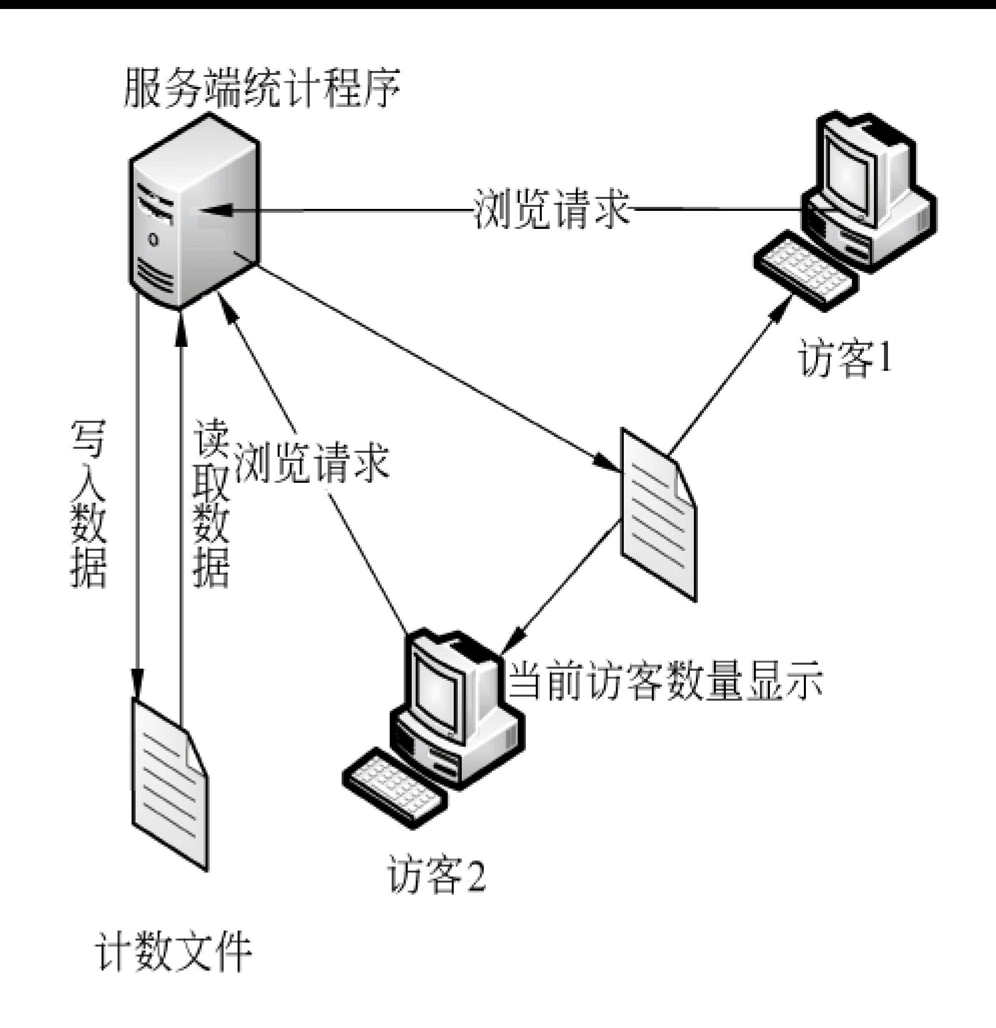


图 4.1　访客计数器工作流程图

4.2.2　创建计数器函数

由于 PHP 本身并没有提供计数用的函数，所以需要编写一个计数器函数。首先来了解 PHP 函数是如何构成的，以及如何创建自己的函数。

PHP 函数的基本构成：

- 在自定的函数名称前要加入 function 的保留字，表明是用户自定义的函数，并以 function()作为开始。
- 为函数命名。函数名尽量体现出该函数的主要功能，同时方便记忆。
- 以{符号作为函数的开始标记，之后的部分是函数的代码。
- 函数功能主体部分。
- 以}符号作为函数结束的标记。

注意：函数名是以任何英文字母开头的字符串。字符串除了开头不能是数字或下划线外，在第一个字母后可以是数字或者下划线，当然其他的符号或是中文字不能当函数名。

下面是一个简单的例子，该自定义函数在被调用时会输出 Hello。

```
<?php
    function printword(){
    echo "Hello";
    }
    printword();
?>
```

计数器函数的设计步骤如下：

（1）创建一个 function.php 文件，该文件用于存放访客计数器相关的函数；

（2）创建 counter()函数，该函数的功能是创建一个用于记录访问数量的 txt 文本，每次调用该函数的时候获得文本文件中的访问数并记录本次访问返回记录数。

具体的编码实现如下：

```
<?php
    function counter(){             //访客计数器函数
    $file='counter.txt';            //访问数量记录文件
    if(!file_exists($file))         //判断是否为初次登录并创建记录文件
    {
        $num=0;
        $cf=fopen($file,"w");       //打开一个本地文件
        fwrite ($cf,$num);          //初始化计数器
        fclose($cf);                //关闭文件
        }else{
    $cf=fopen($file,"rw");          //获得当前访问数
    $num=fgetc($cf);
    fclose($cf);
    }
                                    //计数器记录本次访问
    $num++;
    $cf=fopen($file,"w");           //更新访问数
    fwrite ($cf,$num);
    fclose($cf);
    return $num;                    //返回当前访问数
}
?>
```

注意：其中$cf 为创建文件的句柄，用于打开计数器文件。

下面对 counter()函数中涉及的 PHP 系统函数做简要说明，见表 4.1。

表 4.1　常用文件操作函数表

函　数　名	说　　明
file_exists	检查文件或目录是否存在，存在则返回 true，否则返回 false
fopen	打开文件或者 URL
fwrite	写入文件（可安全用于二进制文件）
fgets	从文件指针中读取一行
fclose	将handle指向的文件关闭.如果成功则返回true，失败则返回false

4.2.3　计数器完整实现

在完成了计数器函数之后，可以调用这个自定义的函数获得当前的访问数据，同时可以根据不同用途灵活地显示数据内容信息。下面以文字信息的展现方式作为示例，详细说明计数器的编程过程。

1．文字信息方式的访客计数器

由于 funciton.php 文件已经封装了 counter()计数器函数，所以下面将创建一个简单的访问页面，该文件命名为 counter.php，创建该文件后将函数嵌入访问页面中。具体代码实现如下：

```
<?php
//加载自定义函数文件
```

```
require("funciton.php");
?>
<HTML>
    <HEAD>
        <TITLE>访客计数器</TITLE>
    </HEAD>
<BODY>
    <CENTER>
        <H1>欢迎访问</H1>
            <BR>
                <HR>
                    <FONT SIZE= 7 COLOR= red>
                        //嵌入计数器
                        您是第<?php echo counter();?>位访客
                    </FONT>
    </CENTER>
</BODY>
</HTML>
```

在浏览器中运行 counter.php，效果如图 4.2 所示。同时在当前目录下会生成一个名为 counter.txt 的文件，该文件就是存储计数结果值的文本文件。



图 4.2　文本形式访客计数器运行效果图

4.3　组合条件查询例子

在网站开发中，经常要用到在各种组合条件下的结果检索，用户按照自己的需要获取有效信息。在技术层面主要涉及了 PHP 与 MySQL 数据库的结合，以及 SQL 语句在 PHP 的具体实现。本节将以一个房屋装修查询的例子，详细介绍 PHP 是如何结合数据库完成组合条件查询的。

4.3.1　组合条件查询系统构架

本节介绍的查询系统是一个很典型的组合条件查询的例子，它具有普遍性的意义。读者在了解整个流程后，可以根据不同的业务定制不同的系统，其实现原理都是一样的。图 4.3 展示了组合条件查询的工作流程。

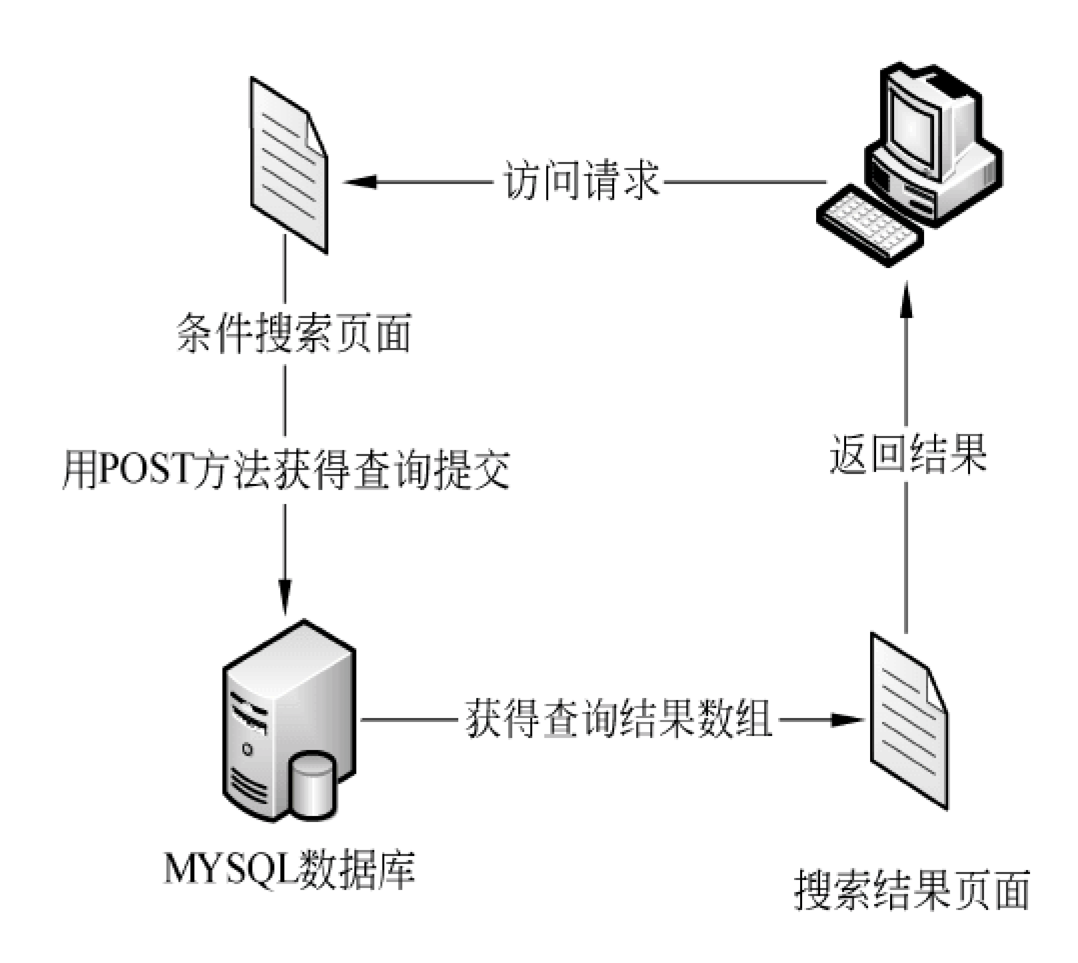


图 4.3　组合条件查询的工作流程

4.3.2　数据库设计

该组合条件查询程序是以一个房屋装修查询为业务模型设计的，由于该业务逻辑比较简单，这里只涉及一个表（case 表），即装修案例表。表 4.2 详细展示了所用到的字段及对应含义。

表 4.2　case表字段及含义

字　段　名	含　　义
pid	装修方案编号，系统设定为自增
idea	装修理念。如朴素设计、实用简约
housetype	房屋类型，如二室一厅、三室二厅
style	装修风格、田园风格、中式风格
app	房屋用途，如自住、商铺
price	预计装修费用
pubdate	发布日期

建立一个新的数据库命名为 search，MySQL 字符集，即 UTF-8 Unicode (utf8)。使用如下 SQL 语句建立 case 表并向表中导入测试用的数据。

```
-- phpMyAdmin SQL Dump
-- version 2.10.2
-- http://www.phpmyadmin.net
--
-- 主机: localhost
-- 生成日期: 2010 年 02 月 17 日 06:11
-- 服务器版本: 5.0.45
-- PHP 版本: 5.2.3

SET SQL MODE="NO AUTO VALUE ON ZERO";

--
-- 数据库: 'search'
--
```

```
-- --------------------------------------------------------

--
-- 表的结构 'case'
--

CREATE TABLE 'case' (
  'pid' mediumint(9) NOT NULL auto increment,
  'idea' varchar(128) NOT NULL,
  'housetype' varchar(128) NOT NULL,
  'style' varchar(128) NOT NULL,
  'app' varchar(64) NOT NULL,
  'price' varchar(64) NOT NULL,
  'area' float NOT NULL,
  'pubdate' datetime NOT NULL,
  PRIMARY KEY  ('pid')
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO INCREMENT=7 ;

--
-- 导出表中的数据 'case'
--

INSERT INTO 'case' VALUES (1, '实用简约', '二室一厅', '田园风格', '自住', '40000',
80, '2010-02-17 13:23:17');
INSERT INTO 'case' VALUES (2, '朴素设计', '二室一厅', '中式风格', '自住', '45000',
100, '2010-02-09 13:25:40');
INSERT INTO 'case' VALUES (3, '实用简约', '三室二厅', '田园风格', '自住', '60000',
120, '2010-02-02 13:29:06');
INSERT INTO 'case' VALUES (4, '朴素设计', '三室二厅', '中式风格', '自住', '65000',
150, '2010-02-03 13:29:13');
INSERT INTO 'case' VALUES (5, '实用简约', '三室二厅', '田园风格', '自住', '60000',
120, '2010-02-02 13:29:06');
INSERT INTO 'case' VALUES (6, '实用简约', '三室二厅', '田园风格', '自住', '65000',
150, '2010-02-03 13:29:13');
```

4.3.3　条件查询实例的功能规划

组合条件查询的例子用到如下文件及页面，这些文件的具体功能规划如下所示。

- 条件搜索页面（index.html）：负责收集用户选择的条件信息，并通过表单形式发送给程序处理。
- 结果处理及展示文件（search.php）：根据从表单得到的筛选条件生成 SQL 语句，并交由 MySQL 处理生成结果数组并返回给用户。

4.3.4　条件搜索页面

建立一个 HTML 文件，命名为 index.html。该文件包含一些组合查询所需要的查询，如房屋类型、装修风格等。该文件具体内容如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>条件搜索页面</title>
```

```
</head>
<form action="search.php" method="post">
<body>
<table width="80%" border="1" align="center" cellpadding="2" bordercolor=
"#0000FF" bordercolorlight="#7D7DFF" bordercolordark="#0000A0" bgcolor=
"#E8E8E8">
<!--装修理念下拉菜单-->
<tr>
    <td width="12%" height="29" align="left" valign="middle">装修理念</td>
        <td width="88%" height="29" align="left">
            <select name="idea" id="select1">
                <option value="实用简约">实用简约</option>
                <option value="朴素设计">朴素设计</option>
            </select>
        </td>
</tr>
<!--房屋类型下拉菜单-->
<tr>
   <td width="12%" height="29" align="left" valign="middle">房屋类型</td>
        <td width="88%" height="29" align="left">
            <select name="housetype" id="select2">
                <option value="二室一厅">二室一厅</option>
                <option value="三室二厅">三室二厅</option>
            </select>
        </td>
</tr>
<!--装修风格下拉菜单-->
<tr>
    <td width="12%" height="29" align="left" valign="middle">装修风格</td>
        <td width="88%" height="29" align="left">
            <select name="style" id="select3">
                <option value="田园风格">田园风格</option>
                <option value="中式风格">中式风格</option>
            </select>
    </td>
</tr>
<!--房屋用途下拉菜单-->
<tr>
    <td width="12%" height="29" align="left" valign="middle">房屋用途</td>
        <td width="88%" height="29" align="left">
            <select name="app" id="select4">
                <option value="自住">自住</option>
                <option value="商铺">商铺</option>
            </select>
    </td>
</tr>
<!--期望价格下拉菜单-->
<tr>
    <td width="12%" height="29" align="left" valign="middle">期望价格</td>
        <td width="88%" height="29" align="left">
            <select name="price" id="select5">
                <option value="A">50,000 以下</option>
                <option value="B">50,000 以上</option>
            </select>
    </td>
</tr>
<!--发布时间下拉菜单-->
<tr>
```

```
    <td width="12%" height="29" align="left" valign="middle">发布时间</td>
        <td width="88%" height="29" align="left">
            <select name="pubdate" id="select6">
                <option value="A">最近一周</option>
                <option value="B">最近一个月</option>
                <option value="C">最近三个月</option>
                <option value="D">最近半年</option>
            </select>
    </td>
</tr>
     <tr>
        <td> </td>
            <td>
                <input type="submit" value="提交">
            </td>
</tr>
</form>
</table>
</body>
</html>
```

运行上述代码，效果如图 4.4 所示。

图 4.4　条件搜索页面运行效果图

4.3.5　结果处理及展示文件

建立一个 PHP 文件，该文件命名为 search.php，该文件的作用是结果处理及展示。根据从表单得到的筛选条件生成 SQL 语句，交由 MySQL 处理生成结果数组并返回给用户。这部分通过将 PHP 作为脚本直接嵌入到 HTML 代码中，作为结果展示页面。

文件 search.php 代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>查询结果页面</title>
</head>
<body>
<table align="center">
<?
```

```
    $idea=$_POST['idea'];                           //获得装修理念
    $housetype=$_POST['housetype'];                 //获得房屋类型
    $style=$_POST['style'];                         //获得装修风格
    $app=$_POST['app'];                             //获得房屋用途
    $price=$_POST['price'];                         //获得期望价格
    $pubdate=$_POST['pubdate'];                     //获得发布日期
    $conn=mysql_connect("localhost","user","password");    //创建数据库链接
    mysql_select_db("search");                      //选择数据库
    $query="select * from 'case' where 'idea'='".$idea."'";
                                                    //通过"组合"的方式生成查询用的 SQL
    if($housetype!=""){                             //判断房屋类型是否被选择
        $query.=" && 'housetype'='".$housetype."'";
    }
    if($style!=""){                                 //判断装修风格是否被选择
        $query.=" && 'style'='".$style."'";
    }
    if($app!=""){                                   //判断房屋用途是否被选择
        $query.=" && 'app'='".$app."'";
    }
    if($price!=""){                                 //判断期望价格是否被选择
        switch($price){
            case "A":
                $query.=" && 'price'<'50000'";
                break;
            case "B":
                $query.=" && 'price'>='50000'";
                break;
    }
}
switch($pubdate){                                   //判断信息发布日期
    case "A":
        $query.=" && TO_DAYS(NOW()) - TO_DAYS('pubdate')<=7";//最近一周发布
        break;
    case "B":
        $query.=" && TO_DAYS(NOW()) - TO_DAYS('pubdate')<=30";
                                                    //最近一个月发布
        break;
    case "C":
        $query.=" && TO_DAYS(NOW()) - TO_DAYS('pubdate')<=91";
                                                    //最近三个月发布
        break;
    case "D":
        $query.=" && TO_DAYS(NOW()) - TO_DAYS('pubdate')<=183";
                                                    //最近半年发布
        break;
}
$query=$query." ORDER BY 'pid' limit 0,20 ;";  //排序
//由于数据字符集为 UTF-8 通过下面一条语句将统一字符问题输入输出
@mysql_query("SET NAMES 'utf8'",$conn);
$result=mysql_query($query,$conn);                  //获得查询结果
if($result){                                        //判断是否成功获得查询结果
    $rows=mysql_num_rows($result);                  //遍历数组内容
        if($rows!=0){
        while($myrow=mysql_fetch_array($result)){
            echo "<tr>";
            echo "<td width='15' height='12'><input name='t1' type='radio'
            value='' /></td>";
            echo "<td width='540' height='12'>$myrow[pid] 
```

```
            $myrow[idea]  $myrow[style]($myrow[housetype])
            <font style='font-size:9pt'>[$myrow[pubdate]]</font>";
            echo "</td>";
            echo "<td width='75' height='12'><a href='' target='_blank'>
            详细信息</a></td>";
            echo "</tr>";
        }
    }
    }else{
    echo "<tr><td><div align='center'><br><br>没有找到满足你条件的记录</div>
    </td></tr>";
}
echo "<hr width=\"100%\" size=\"1\">";
?>
</table>
</body>
</html>
```

注意：字符集一直困扰着很多开发者，尤其在中文和英文混合的页面处理起来问题非常突出。笔者也曾遇到类似的问题，推荐解决办法为 MySQL 字符集采用 UTF-8 Unicode (utf8)，HTML 页面也采用 UTF-8 编码在 PHP 通过@mysql_query("SET NAMES 'utf8'",$conn);来达到统一字符集的目标。

程序运行后的效果如图 4.5 所示。

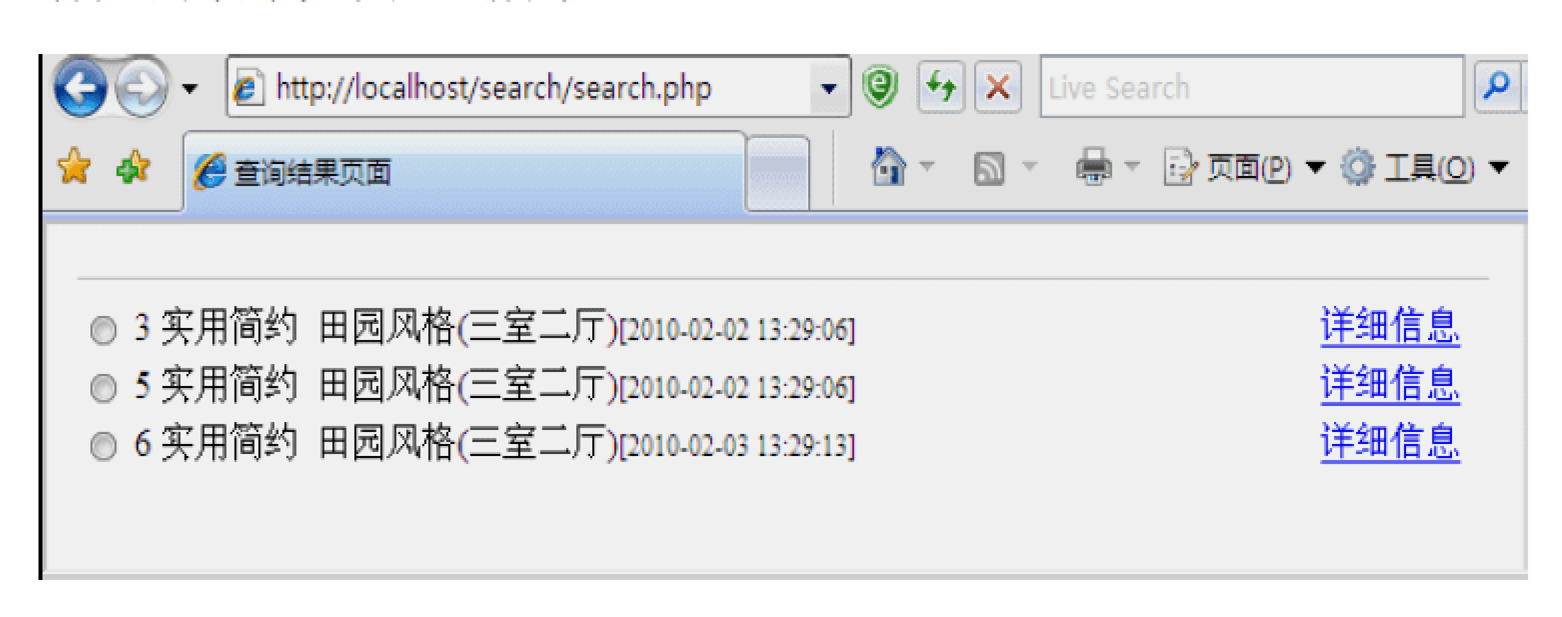


图 4.5　查询结果图

4.4　小　　结

本章从一个简单的入门例子 Hello World 开始，讲述了 PHP 开发的基础知识。第 4.2 节介绍了“访客计数器”的程序实现，这部分结合实际应用介绍了 PHP 是如何实现文件存取的。第 4.3 节介绍了“组合条件查询”，通过一个装修的例子着重讨论了 PHP 与 MySQL 数据库的结合。以上的 3 个部分也代表了 PHP 开发基本的应用形式。本章的 PHP 示例是学习 PHP 语言的经典入门示例，读者在学习本章时需要着重理解每个示例代表的技术特点。

第 2 篇　PHP 典型模块开发与应用

- ⏭ 第 5 章　留言板模块（表单+Session+验证+分页）
- ⏭ 第 6 章　注册及登录验证模块（验证+正则表达式+SQL 注入）
- ⏭ 第 7 章　上传与下载模块（PHP 防盗链+ActiveXObject 控件）
- ⏭ 第 8 章　内容自动采集器模块（爬虫+FCKeditor 编辑器+任务接力）
- ⏭ 第 9 章　日程表模块（PHP+AJAX+XML）
- ⏭ 第 10 章　聊天室模块　（Frameset 页面框架+Session）
- ⏭ 第 11 章　计费系统模块（原子操作+数据回滚 ROLLBACK）
- ⏭ 第 12 章　充值平台模块（MD5 加密+HTTPS 协议）
- ⏭ 第 13 章　在线调查模块（PHP+MySQL+XML）
- ⏭ 第 14 章　WAP 资讯模块（PHP+WML+WAP）
- ⏭ 第 15 章　广告管理模块（JavaScript+iframe+Smarty）
- ⏭ 第 16 章　文件管理器（文件处理+Smarty）

第 5 章　留言板模块（表单+Session+验证+分页）

留言板模块程序本身不是一个特别复杂的应用，但是留言板模块程序涉及了 Web 开发的各个方面，通过本章的介绍读者会对 Web 开发的标准流程有个完整的认识。留言板模块程序的设计和编码实现是一个标准的以需求为向导的开发模式，本章将从基础设计入手，为读者逐一介绍开发步骤，并最终完成留言板模块。

本章主要涉及的知识点如下。

- Session 会话：在整个用户会话过程中，通过将对象存储在 Web 服务器的内存中来保持状态，或者说保存值。
- Form 表单包含有 3 个基本要素：表单控件（Form Controls）、Action 和 Method。
- 表单的提交方式：POST 方式和 GET 方式。
- 表单内容有效性验证：JavaScript 验证、正则表达式匹配验证。
- MySQL 数据库的基本操作：插入（insert）、查询（select）、更新（update）、删除（delete）操作。

5.1　基 础 设 计

留言板的核心功能是提供访客在网站上留言的功能。业务流程的步骤是首先接收访问者输入的信息，将信息存入网站数据库完成留言，访客通过 Web 页面将访客的留言显示出来。通过业务流程的描述可以了解到，留言板模块程序按照业务流程分为提交留言和显示留言两部分。

提交留言功能将数据存入数据库，显示留言功能将数据库中的信息显示于页面上。其中提交留言和留言管理是两个最重要的模块，由于该留言板不涉及用户注册流程，所以简化了管理员登录及权限认证的过程。

5.1.1　功能描述

留言板的核心功能就是为网站运营者与用户提供一个在网站双向交流的内容区域，该区域可记录访客留言、收集意见和建议，为优化服务提供数据依据。核心功能描述如下。

- 留言提交：访客提交留言。
- 留言/回复查看：访客查看留言及管理员回复。
- 管理员身份认证：管理员后台登录并获得管理权限。

- 管理员留言回复：管理员答复访客留言。
- 管理员删除留言：管理员删除访客留言。

根据以上的功能描述，创建以下文件来实现相关功能。

1．留言浏览页

页面名称：index.php（入口文件）。
主要功能描述：按照留言先后顺序，分页显示留言信息。
语言类型：PHP。
涉及的数据表：客户留言表（info）、留言回复信息表（reply）。

2．留言处理页

页面名称：sent.php。
主要功能描述：验证用户输入的数据完整性和合法性，保存用户信息和留言内容。
语言类型：PHP、JavaScript。
涉及的数据表：客户留言表（info）、留言回复信息表（reply）。

3．管理员登录功能详细设计

页面名称：login.php。
主要功能描述：管理员登录程序。
语言类型：PHP、JavaScript。
涉及的数据表：无。

注意：登录验证部分为 PHP 中创建 Session 全局变量来实现登录验证及管理权限获得功能，因此没有用到数据库用户密码表一类的设计。

4．留言回复文件

页面名称：reply.php。
主要功能描述：答复指定访客的留言。
语言类型：PHP、JavaScript。
涉及的数据表：留言回复信息表（reply）。

5．留言删除文件

页面名称：delete.php。
主要功能描述：删除指定的访客留言。
语言类型：PHP 、JavaScript。
涉及的数据表：客户留言表（info）、留言回复信息表（reply）。

5.1.2　流程描述

留言板模块程序涉及的工作流程十分清晰，因此很容易理解，具体流程如图 5.1 所示。

该留言板没有单独提供管理员的管理后台，因此访客留言和管理员的管理操作是在同一个界面完成的，由全局状态变量来区分操作权限，该方法是小型项目中一种简捷的解决方案。其中，管理员的管理权限部分没有单独设计信息表，在登录文件 login.php 采用用户名和密码明文存储比对的方式进行验证，代码如下：

```
//根据表单传入值比对现有明文密码和用户名
if($_POST['username']=='admin'&&$_POST['password'])=='admin'{
    $_SESSION['login']=true;        //创建全局变量即登录状态为 true
    echo "<script>alert('管理员登录成功');location.href='index.
    php';</script>";
    exit();
}
else {
    echo "<script>alert('登录失败！');</script>";
}
```

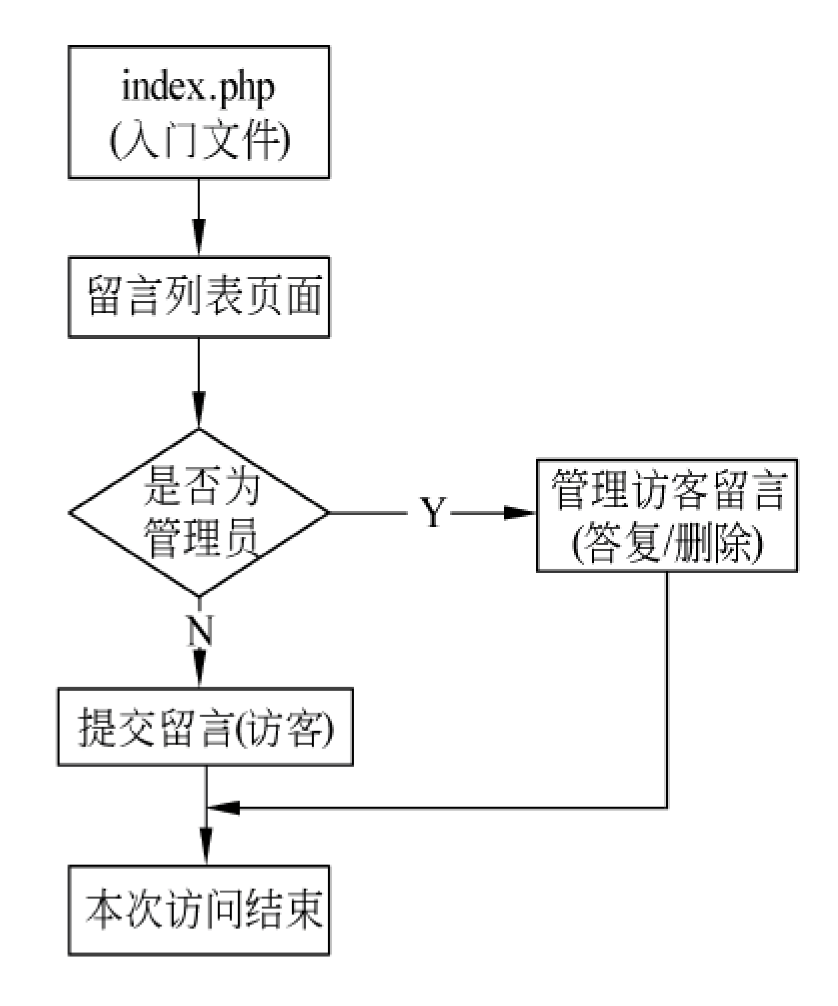


图 5.1　留言板流程图

5.2　数据库设计

根据前面的功能需求描述和流程描述，本例的留言板的数据库“实体-关系”，如图 5.2 所示，其中有两个表：

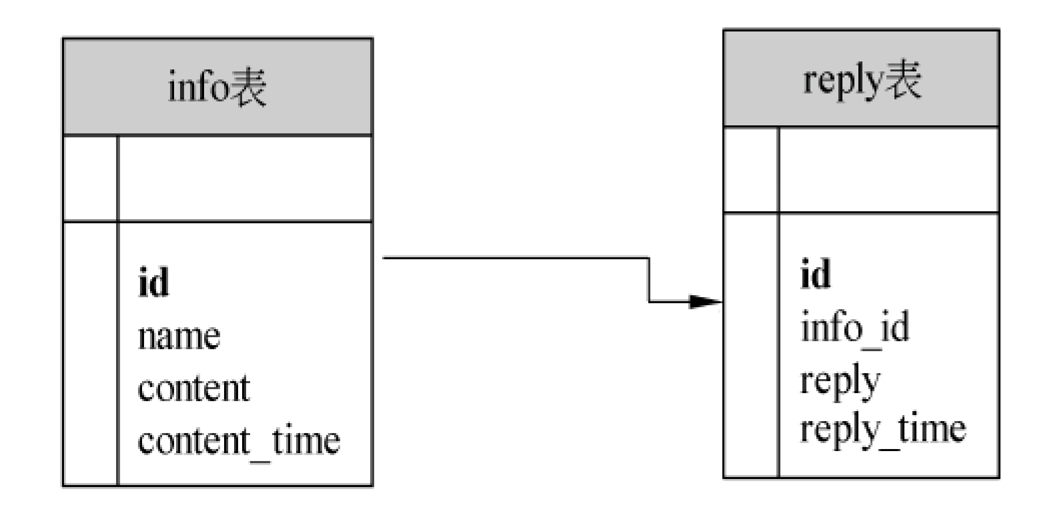


图 5.2　Guestbook 数据库 E-R 关系图

❑ info 表用来存储访客留言和个人资料信息。

- reply 表用来存储管理员回复信息。

这里创建一个数据库 GuestBook，数据库的字符集为“UTF-8 Unicode (utf8)”。创建数据库 SQL 语句如下：

```
CREATE DATABASE 'GuestBook' DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci;
```

5.2.1　访客留言信息表

访客留言信息表（info 表）创建脚本的语句如下：

```
--
-- 表的结构 'info'
--
CREATE TABLE IF NOT EXISTS 'info' (
  'id' int(11) NOT NULL auto increment COMMENT '留言自增 id',
  'name' varchar(16) NOT NULL COMMENT '访客名字',
    'content' text NOT NULL COMMENT '发布内容',
    'content time' varchar(14) NOT NULL COMMENT '发布时间',
  PRIMARY KEY  ('id')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
```

5.2.2　管理员回复留言表

管理员回复留言表（reply 表）创建脚本的语句如下：

```
--
-- 表的结构 'reply'
--
CREATE TABLE IF NOT EXISTS 'reply' (
  'id' int(11) NOT NULL auto_increment COMMENT '自增 id',
  'info_id' varchar(11) NOT NULL COMMENT '留言 id',
  'reply' text NOT NULL COMMENT '回复内容',
  'reply_time' varchar(14) NOT NULL COMMENT '回复时间',
  PRIMARY KEY  ('id')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;
```

5.3　留 言 浏 览

留言浏览部分是整个留言板的核心，同时该页面也是程序的入口文件，用户通过访问留言浏览页面后，才能获得有效的信息继续后续的操作。该页面是普通用户查看留言和查看管理员留言反馈的页面，管理员用户通过该页面查看和管理访客提交的留言信息。

5.3.1　留言浏览页面设计

留言浏览页面部分的设计，按照功能分区可以划分为。

- 访客信息展示：留言人姓名、留言提交时间。

- 留言内容展示：访客提交留言信息。
- 管理员操作区：回复、删除管理功能链接。
- 留言回复区：管理员回复内容、回复时间、回复标题。

下面介绍留言浏览页面，创建一个临时文件 guestbook.html 用于预览页面。页面模板代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>留言浏览模板</title>
<link href="style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<!--中部内容开始-->
<div id="content">
  <div id="main">
    <ul>
      <li id="main top"></li>
      <li id="main middle" style="padding:0px 0px 15px 0px;">
        <div class="title_cr">留言板</div>
          <table width="872" border="0" align="center" cellpadding="0"
          cellspacing="0" class="tab bbs">
          <tr>
            <td colspan="2" class="title_bbs">
                <table width="100%" border="0" cellspacing="0"
                cellpadding="0">
                <tr>
                  <td width="50" class="title b"> </td>
                  <td width="400" class="title_b">
                    <a href="reply.php?id=<?php echo $rs->id?>">回复</a> |
<a href="delete.php?id=<?php echo $rs->id?>">删除</a>
                        建议
                  </td>
                  <td width="100" class="title b">
                    <!--留言人-->
                    [留言人姓名]
                  </td>
                  <td class="title b">
                    <!--留言时间-->
                    [留言提交时间]
                  </td>
                </tr>
              </table></td>
          </tr>
          <tr>
            <td width="45"> </td>
            <td width="823" class="td bbs">
                <!--留言内容开始-->
              [留言内容展示区]
              <!--留言内容结束-->
            </td>
          </tr>
          <tr>
            <td width="45" class="td_bbswhite"> </td>
                <td class="td bbsgray">
回复标题：你好　 回复人：管理员 回复时间：[2010/01/26]
</td>
```

```
          </tr>
          <tr>
            <td width="45" class="td bbs"> </td>
             <td>
<!--回复留言内容开始-->
              [管理员回复留言区]
              <!--回复留言内容结束-->
</td>
          </tr>
        </table>
        <div style="clear:both;"></div>
        <div id="dashline"></div>
        <!--页码开始-->
        <div id="page" style="margin-right:30px; margin-top:8px;">
         <span class="grayborder" style="background-color:#F6F6F6;">
           <a href="4">下一页</a>
         </span>
        ⋮
         <span class="grayborder" style="background-color:#F6F6F6;">
            <a href="5">上一页</a>
         </span>
        </div>
        <!--页码结束-->
      </li>
      <li id="main bt"></li>
    </ul>
  </div>
</div>
<!--中部内容结束-->
</body>
</html>
```

🔔注意：style.css 为该模板的样式配置文件，里面设定了部分页面样式参数。

在浏览器中打开 guestbook.html 文件，预览效果如图 5.3 所示。



图 5.3　留言浏览页面预览图

🔔注意：在实际编写页面代码的时候会遇到浏览器兼容的问题，在不同浏览器下显示的效果会有很大差异。其实这与开人员代码的编写是否严谨有关，也与不同浏览器本身的特性有关。笔者建议以 IE 6.0 浏览器为基础平台进行调试，待通过之后再逐步优化代码兼容其他浏览器。

5.3.2　留言浏览程序设计

留言浏览程序涉及以下两个文件。

- 浏览程序文件（index.php 即入口程序）。
- 数据库配置文件（config.inc.php）。

在数据库配置文件（config.inc.php）中，定义了留言板模块程序相关数据库的配置信息，其内容如下：

```
<?php
    session_start();
    //数据库配置信息
    define('DB_HOST', 'localhost');         //数据库服务器主机地址
    define('DB_USER', 'root');              //数据库账号
    define('DB_PW', 'pwd');                 //数据库密码
    define('DB_NAME', 'GuestBook');         //数据库名
    define('DB_CHARSET', 'utf8');           //数据库字符集
    define('DB_PCONNECT', 0);               //0 或 1，是否使用持久连接
    define('DB_DATABASE', 'mysql');         //数据库类型
    //创建数据库链接句柄
    $con=mysql_connect(DB_HOST,DB_USER,DB_PW) or die('链接数据库失败！');
    mysql_query('set names utf8');          //设定字符集编码
    mysql_select_db(DB_NAME);               //选择数据库
?>
```

留言浏览程序（index.php）作用是将数据库 info 表和 reply 表中对应的信息取出，再通过循环遍历的方式，将结果输出，以得到留言列表和相关匹配的信息。

留言浏览部分代码内容如下：

```
<?php
    require ('config.inc.php');//加载数据库配置文件
    //连表组合查询语句
    $sql = "SELECT c1 . * , c2.reply_time, c2.reply FROM info c1 LEFT JOIN
    reply c2 ON ( c1.id = c2.info_id ) ORDER BY c1.id DESC";
    //获得结果集中的记录数
    $numRecord = mysql_num_rows(mysql_query($sql));
    $recordSql = $sql;
    $result = mysql_query($recordSql);
    //循环遍历结果集中的留言信息
    while($rs=mysql_fetch_object($result)){
        //echo nl2br(htmlspecialchars($rs->content));
        //该部分为输出留言相关的信息
    }
    //释放 MySQL 数据库 query 返回所占用的内存
    mysql_free_result($result);
?>
```

注意：本段程序中 MySQL 的连表查询是这段程序的核心。

PHP 本身提供了丰富的操作 MySQL 的函数，但是对相似功能的函数很多开发者容易混淆。以上部分内容涉及的都是 PHP 中最常用的 MySQL 数据库函数，熟练掌握并了解相

关的函数之后对其他项目的开发很有帮助。表 5.1 为常见 MySQL 数据库函数。

表 5.1　常见MySQL数据库函数

函　数　名	描　　述
mysql_fetch_array($result)	用来将查询结果 result 拆到数组变量中
mysql_fetch_object($result)	功能与 mysql_fetch_array()几乎相同，不同的地方在于本函数返回资料是类而不是数组
mysql_num_rows()	得到结果集中返回列的数目
mysql_num_fields()	可以得到返回字段的数目
mysql_query()	送出 query 字符串供 MySQL 做相关的处理或者执行
mysql_free_result()	释放当前 MySQL 数据库 query 返回所占用的内存

5.3.3　分页处理技术

在 MySQL 数据库中，信息的分页处理是一个最常见的需求，要想了解实现的原理需要从 MySQL 语句本身结构说起，先来看一条最常见的 SQL 查询语句。

```
select * from table limit 0,10
```

其含义是从指定表中取全部字段记录中的前 10 条，通过归纳这条语句就可以得出如下语句：

```
select * from table limit $offset,$num
```

其中，$offset 表示传入的页面数–1。$num 是每个页面显示的行数，通常在配置文件中统一进行设置。在程序文件中加入处理翻页的代码，并将程序处理代码整合到之前设计的“留言浏览模板”中重新命名为 index.php 文件。该文件完成了“留言浏览”部分的完整内容，改动后的 index.php 代码如下：

```
<?php
    require ('config.inc.php');                    //加载数据库配置文件
    $pagesize = 5;                                 //设定每页显示条目数
    if(isset($_GET['page'])&&$_GET['page']!='') {
        $page=$_GET['page'];
        }else {
        $page=0;
    }
    //连表组合查询 SQL 语句
    $sql = "SELECT c1 . * , c2.reply_time, c2.reply FROM info c1 LEFT JOIN
    reply c2 ON ( c1.id = c2.info_id ) ORDER BY c1.id DESC";
    $numRecord = mysql_num_rows(mysql_query($sql));//获得结果集中的记录数
    $totalpage = ceil($numRecord/$pagesize);       //获得总页数
    $recordSql = $sql. " LIMIT ".$page*$pagesize.",".$pagesize;
                                                   //拼接翻页 SQL 语句
    $result = mysql_query($recordSql);
?>
    <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>留言板</title>
    <link href="style.css" rel="stylesheet" type="text/css" />
```

```
</head>
<body>
<!--中部内容开始-->
<div id="content">
<div id="main">
<ul>
    <li id="main top"></li>
    <li id="main middle" style="padding: 0px 0px 15px 0px;">
    <div class="title_cr">留言板</div>
    <?php
//循环嵌套开始
while($rs=mysql fetch object($result)){
?>
    <table width="872" border="0" align="center" cellpadding="0"
        cellspacing="0" class="tab bbs">
        <tr>
            <td colspan="2" class="title bbs">
            <table width="100%" border="0" cellspacing="0" cellpadding=
            "0">
                <tr>
                    <td width="50" class="title b"> </td>
                    <td width="400" class="title b"><a href="reply.
                    php?id=<?php
                    echo $rs->id?>">回复</a> | <a href="delete.
                    php?id=<?php echo
                    $rs->id?>">删除</a>     建议</td>
                    <td width="100" class="title_b"><!--留言人--> <?php
                    echo $rs->name?>
                    <!--留言人--></td>
                    <td class="title b"><!--留言时间--> <?php echo date
                    ("Y-m-d H:i:s",$rs->content_time+8*3600)?>
                    <!--留言时间--></td>
                </tr>
            </table>
            </td>
        </tr>
        <tr>
            <td width="45"> </td>
            <td width="823" class="td bbs"><!--留言内容--> <?php echo
            nl2br(htmlspecialchars($rs->content))?>
            <!--留言内容--></td>
        </tr>
        <tr>
            <td width="45" class="td_bbswhite"> </td>
            <td class="td bbsgray">回复标题：你好 回复人：管理员 回复时间：
            <?php if($rs->reply_time!="") echo date("Y-m-d H:i:s",
            $rs->reply_time+8*3600)?></td>
        </tr>
        <tr>
            <td width="45" class="td bbs"> </td>
            <td><!--回复留言内容--> <?php echo nl2br(htmlspecialchars
            ($rs->reply))?>
            <!--回复留言内容--></td>
        </tr>
    </table>
    <?php
//循环嵌套收尾
}
?>
```

```
        <div style="clear: both;"></div>
        <div id="dashline"></div>
        <!--页码开始-->
        <div id="page" style="margin-right: 30px; margin-top: 8px;"><span
            class="grayborder" style="background-color: #F6F6F6;"> <a
            href='index.php?page=<?php if ($page<$totalpage-1) echo
        $page+1;?>'>下一页</a> </span> <span class="grayborder"
            style="background-color: #F6F6F6;"> <a href='index.php?page=
            <?php
        if ($page>0) echo $page-1;?>'>上一页</a> </span></div>
        <!--页码结束--></li>
        <li id="main_bt"></li>
    </ul>
    </div>
```

【代码解读】

上述程序首先加载基础配置文件、配置变量参数，然后通过“连表组合查询 SQL 语句”获得结果集中的记录数和获得总页数，最后拼接翻页 SQL 语句并输出到页面模板中。

5.4 留 言 提 交

关于留言提交功能读者并不陌生，几乎所有的网站都有类似的功能设计。其目的就是增加网站的互动性，建立一个和访问者交流的渠道，提升服务质量、改进产品设计。

在技术层面留言提交主要涉及以下内容。

- 表单内容有效性验证：JavaScript 验证、正则表达式。
- 留言内容处理：信息完整性检查。

5.4.1 留言提交页面设计

留言提交页面的核心就是在表单的设计上创建“post.html”文件，用于调试和预览留言提交页面。留言提交页面（post.html）的文件代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>留言板</title>
<link href="style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="mainbt">
<ul>
    <li id="mainbt top"></li>
    <li id="mainbt middle" style="padding: 0px 0px 15px 0px;">
    <p class="tit_bbs">我要留言</p>
    <!--表单起始，form 标签内的内容将会被提交-->
    <form action="sent.php" method="post" name="message board">
    <table width="80%" border="0" align="center" cellpadding="0"
        cellspacing="0" class="tab b">
        <tr>
            <td class="td_bright">用户名：</td>
```

```
            <td width="150" class="td pad"><label> <input
                type="text" name="name" id="name" /> </label></td>
            <td width="50">验证码：</td>
            <td width="90" class="td pad"><label> <input type="text"
                name="password" style="width: 80px;" /> </label></td>
            <td><img src="images/checkno.gif" /></td>
        </tr>
        <tr>
            <td class="td_bright">提交部门：</td>
            <td colspan="4" class="td pad"><label> <select
                name="select">
                <option value="#">—— 全部 ——</option>
            </select> </label></td>
        </tr>
        <tr>
            <td valign="top" class="td bright">发表内容：</td>
            <td colspan="4" class="td_pad">
            <label>
            <textareaname="content" id="post" cols="40" rows="5"></textarea>
            </label>
            </td>
        </tr>
        <tr>
            <td> </td>
            <td colspan="4">
            <table width="20%" border="0" cellspacing="0" cellpadding="0">
                <tr>
                    <td class="td pad"><input type="submit" name="Submit"
                        value="提交" onclick="return checkInput();" /></td>
                    <td class="td_pad"><input type="reset" name="Submit2"
                        value="重置" /></td>
                </tr>
            </table>
            </td>
        </tr>
    </table>
    </form>
    </li>
    <li id="mainbt_bt"></li>
</ul>
</div>
</body>
</html>
```

【代码解读】

在 1.2.5 节 HTML 表单设计一节已经对 Form 标签做了详细描述，在 post.html 中 form 标签如下：

```
<form action="sent.php" method="post" name="message board">
</form>
```

Form 表单包含以下 3 个基本要素。

- 表单控件（Form Controls）：post.html 用到的表单控件有 input type="text"、input type="submit"、input type="reset"、textArea（多行文本输入框）。
- Action：提交到需处理表单信息的文件即 sent.php。
- Method：提交模式有 POST 和 GET 两种方式。

在浏览器中加载 post.html 文件，预览效果如图 5.4 所示。



图 5.4　提交留言页面

5.4.2　留言有效性验证

留言有效性的验证在本例包含两个部分，即用户名的验证和留言内容的验证。具体需求如下所示。

- 验证用户名：限制输入 10 个字符（5 个汉字），不能输入非法字符，不能为空。
- 验证留言内容：留言内容不能为空，留言内容不能过长（不超过 400 字）。

下面介绍留言有效性的验证代码，创建一个 JavaScript 文件用于表单输入验证，命名为 check.js。内容如下：

```
<script language="JavaScript">
function checkInput(){
    var name = document.getElementById('name');
    var post = document.getElementById('post');
    //验证用户名：不能超过 10 个字符（5 个汉字），不能输入非法字符，不能为空
    nameValue = name.value.replace(/\s+/g,"");
    var SPECIAL_STR = "~!%^&*();\"?><[]{}\\|,:/=+—";
    var nameflag=true;
    for(i=0;i<nameValue.lenght;i++){
        if (SPECIAL STR.indexOf(nameValue.charAt(i)) !=-1)
        nameflag=false;
    }
    if(nameValue==''){
        alert('请填写用户名称！');
        return false;
    }
    if(nameValue.length>10){
        alert('用户名称最多 10 个字符（5 个汉字）！');
        return false;
    }
    if(nameflag===false){
        alert('用户名称不能包含非法字符请更改！');
        return false;
    }
    //留言内容验证
    if(post.value==""){
```

```
        alert('请输入留言内容！');
        return false;
    }
    if(post.value.length>400){
        alert('留言内容太长！');
        return false;
    }
}
</script>
```

在留言提交页面 post.html 文件的<head></head>标签中，引入 JS 文件，代码如下：

```
<script src="./check.js"></script>
```

5.4.3　留言内容处理

在完成了提交页面和表单验证 JavaScript 文件之后，下面将留言的信息保存到数据库中，创建一个名称为 sent.php 的脚本文件，用来处理表单提交过来的留言信息。该文件需要完成的功能有两个：

- ❑ 效验表单提交的信息。
- ❑ 按照规则写入数据库。

sent.php 文件代码并不复杂，核心代码就是一条向数据库中写入数据的 SQL 语句，下面是 sent.php 的详细代码：

```
<?php
    header('content-type:text/html;charset=utf-8');
    $name = $_POST['name'];                    //接收表单传入的 name 值
    $content = $_POST['content'];              //接收表单传入的 content 值
        if($name==""||strlen($name)>10){       //用户名长度判断
        echo <<<tem
        <script language="JavaScript">
        alert('请输入正确的有户名');
        history.go(-1);
        </script>
        tem;
    exit();
    }
    if(strlen($content)>400){                  //留言内容长度判断
        echo <<<tem
        <script>
        alert("输入的留言内容太长！");
        history.go(-1);
        </script>
        tem;
    exit();
    }
    require ('config.inc.php');                //加载数据库配置文件
    $content_time = time();                    //获得当前系统时间
    //把留言信息插入 info 表
    $insertSql="insert into info (name,content,content_time) values
    ('$name','$content','$content_time')";
    if(mysql_query($insertSql)){
        echo <<<tem
        <script>
```

```
        alert("留言成功");
        location.href="index.php";          //浏览器页面跳转
        </script>
    tem;
    }
    else{
        echo <<<tem
        <script>
        alert("留言失败");
        location.href="index.php";          //浏览器页面跳转
        </script>
        tem;
    }
?>
```

5.5　留 言 管 理

留言管理需要完成对留言的管理功能。留言管理部分的设计原则是要直观地显示访客，同时便于管理员管理访客留言。由于该留言板模块程序并不涉及复杂的业务流程，所以留言管理部分只提供留言管理中的一些基础管理功能，即管理员登录（获得管理权限）、留言回复、留言删除。读者在了解了实现原理之后，可以根据实际的业务应用来扩展、添加其他管理功能。

5.5.1　管理员登录

管理员登录界面包含 3 个表单控件，分别是用户名输入控件、密码输入控件和表单提交控件，该界面的代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>管理员登录</title>
</head>
<body>
<table>
    <tr>
        <td>
        <form action="login.php" method="POST" name="">用户名:
         <input type="text" name="username" size="20" /> 密码:
         <input type="password" name="password" size="20">
         <input type="submit" value="登录"name="Submit" />
         </form>
        </td>
    </tr>
</table>
</body>
</html>
```

【代码解读】

以上是登录界面的页面代码，管理员登录的目的是为了通过身份认证从而获得管理权限。权限认证的实现方式有很多种，本例将采取创建 Session 全局变量的方式来实现获得

管理权限。具体的实现过程为程序首先比对表单提交的用户名和密码，在认证之后通过如下代码创建$_SESSION['login']变量并赋值为 true。其他涉及与管理相关的文件都首先判断这个全局变量的状态，如果不存在该全局变量或者变量状态不为 true，则不允许执行后面的代码。具体的代码如下：

```
<?php
    //文件名：login.php
    session_start();
    if(isset($_POST['Submit'])){            //检查是否设置 Submit 变量
    if(!get_magic_quotes_gpc()){
        foreach ($_POST as &$items){        //循环遍历 items
            $items = addslashes($items);
        }
    }
    if($_POST['username']=='admin'&&$_POST['password']=='admin'){
        $_SESSION['login']=true;            //设置登录成功标识
        echo "<script>alert('管理员登录成功');location.href='index.php';
        </script>";
        exit();
    }
    else {
        echo "<script>alert('登录失败！');</script>";
    }
}
?>
```

5.5.2　留言回复

留言回复管理中为了便于管理员更直观地操作，提高回复效率，并没有采用管理后台的方式对留言进行管理。在留言浏览页面直接通过连接进行管理操作。创建一个 reply.php 文件用户回复留言，该文件代码如下：

```
<?php
    session_start();
    require ('config.inc.php');         //加载数据库配置文件
    if(!$_SESSION['login']){            //判断是否登录
        echo "<script>alert('没有登录不能回复!');location.href=
        'index.php';</script>";
        exit();
    }
    if($_POST['Submit']){               //判断是否发送
        if(!get_magic_quotes_gpc()){
            foreach ($_POST as $items){
                $items = addslashes($items);
            }
        }
    //当回复内容过长时返回上一步操作
    if(strlen($_POST['reply'])>400){
        echo "<script>alert('回复内容过长!');history.go(-1);</script>";
        exit();
    }
    $info_id = $_POST['info_id'];       //获得操作留言对应的 id
    $reply = $_POST['reply'];           //回复留言内容
    $time = time();                     //获得系统时间
```

```
    $insertRevertSql = "insert into reply (info_id,reply,reply_time)
    value('$info_id','$reply','".$time."')";
    echo $insertRevertSql;
    if(mysql_query($insertRevertSql)){
        echo "<script>alert('回复成功');location.href='index.php';
        </script>";
        exit();
    }
    else {
        echo "<script>alert('回复失败!');history.go(-1);</script>";
    }
}
?>
    <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>无标题文档</title>
    </head>
    <body>
    <table>
        <tr>
            <td>回复内容：</td>
        </tr>
        <tr>
            <td>
            <form action="reply.php" method="POST" name="reply">
            <textarea name="reply" cols="30" rows="5" id="reply">
            </textarea>
            <input type="hidden" name="info_id" value="<?php echo
            $_GET['id']?> "size="20">
            </form>
            </td>
        </tr>
        <tr>
            <td>
            <input type="submit" value="回 复" name="Submit" />
            <input type="button" onclick="JavaScript:history.go(-1);"
            value="放弃" />
            </td>
        </tr>
    </table>
    </body>
    </html>
```

以管理员身份登录后，选择需要回复的留言后将会跳转到 reply.php。运行效果如图 5.5 所示，留言成功后会返回留言浏览页，方便进行其他操作。

图 5.5　留言回复页面

5.5.3　留言删除

网站留言管理的一个重要工作就是留言删除。为了便于管理员删除无效信息，在本例的留言板设计中，将一条访客和管理员的回复看作是一条留言记录。在删除访客留言的同时与该留言相关的一切信息也都会被一同删除，比如当访客留言被删除时，访客的姓名信息及历史上管理员对该留言的回复也会被一并删除。在技术实现方面的删除功能涉及两个表，即 info 表和 reply 表，在执行 SQL 删除语句时需要注意异常情况的处理。

下面介绍删除留言程序的编码。创建删除留言程序文件，命名为 delete.php，代码如下：

```
<?php
    session start();
    header('content-type:text/html;charset=utf-8');
    require ('config.inc.php');    //加载数据库配置文件
    if(!$_SESSION['login']){       //权限验证
        echo "<script>alert('权限不足!');location.href='index.php';
        </script>";
        exit();
    }
    if(isset($ GET['id'])&&$ GET['id']!=""){
        //删除制定 id 在留言回复表中信息
        $delRevertSql="delete from reply where info_id=".$_GET['id'];
        mysql query($delRevertSql);
        //删除制定 id 在留言信息表中信息
        $delGuestSql="delete from info where id = ".$ GET['id'];;
        mysql_query($delGuestSql);
        if(mysql error()==""){
        echo "<script>alert('删除成功!');location.href='index.php';
        </script>";
    }
}
?>
```

【代码解读】

上述代码首先向浏览器声明文件类型和字符集编码，然后检查待删除的留言 id，确认变量存在后，调用删除指定数据的 SQL 语句，完成留言删除操作。

5.6　小　　结

留言板模块是一个相对独立的模块程序，第 5.1 节介绍留言板的基础设计，包括功能描述、流程描述。第 5.2 节数据库设计部分，介绍留言板相关的两个基础表的设计和说明。从第 5.3 节开始进入具体的业务流程编码，首先从留言浏览开始介绍留言浏览页面设计、留言浏览程序设计和分页的处理。第 5.4 节介绍留言提交功能的实现，本节的重点内容是了解留言有效性验证的基本方法。第 5.5 节留言管理部分介绍了留言板管理的几个基本功能，即管理员登录、留言回复、留言删除。留言板模块是网站的一个典型应用，具有普遍性的意义，读者可以掌握设计的基本思路和方法来搭建类似的应用。

第 6 章　注册及登录验证模块（验证+正则表达式+SQL 注入）

现在网站的许多功能都是开放给注册用户浏览和使用的，因此用户的注册及登录验证功能是网站最常使用的功能。由于这部分代码复用率很高，因此本章将用户注册及登录验证程序封装成通用的程序模块，便于在各个平台中使用。注册及登录验证程序通过数据库来记录用户信息，并提供新用户注册及注册后登录的接口。通过本章的学习，读者将掌握创建注册及登录模块程序的基本原理，同时了解用户的登录和注册，及其相关的验证程序开发流程。

本章主要涉及的知识点如下所示。

- PHP 如何获取用户填写的信息。
- 如何使用 PHP 往数据库添加记录及如何在数据库中查找记录。
- 如何使用 JavaScript 脚本语言在客户端编程。
- 如何使用正则表达式进行数据验证。
- PHP 如何使用 Session 来记住用户的登录信息。

6.1　用户信息表设计

注册及登录验证模块的核心是“用户信息表”的设计，该表用来存储用户的基本数据，包括用户名、密码、邮箱地址（通常是用来在密码丢失时取回密码）。由于涉及密码取回等问题，因此用户信息表的作用十分重要。下面来看具体的表结构的设计，首先建立注册用户数据库，操作步骤如下：

（1）建立 register 数据库，SQL 语句如下：

```
CREATE DATABASE 'register' ;
```

（2）在 register 数据库中建立一个命名为 user 的“用户信息表”，如表 6.1 所示。创建表及相关字段 SQL 语句如下：

```
CREATE TABLE 'user' (
'id' INT( 11 ) NOT NULL AUTO INCREMENT PRIMARY KEY ,
'username' VARCHAR( 20 ) NOT NULL ,
'pwd' VARCHAR( 64 ) NOT NULL ,
'email' VARCHAR( 50 ) NOT NULL ,
'lasttime' DATETIME NOT NULL ,
'regtime' DATE NOT NULL
) ENGINE = MYISAM ;
```

表 6.1　用户信息表

字　段	数据类型	长　度	NULL	默认值	字段说明
id	int	11	否		自增主键
username	varchar	20	否	无	用户名
pwd	varchar	64	否	无	用户密码
email	varchar	50	否	无	邮件地址
lasttime	datetime		否		上次登录时间
regtime	date		否		注册时间

由于本章着重讲解用户注册及登录验证的流程，所在数据库设计时只涉及了最基本的功能，读者可以在今后的项目中添加扩展表，扩展表的功能如下。

- 功能表：主要存储系统用户的操作功能。
- 角色表：存储系统使用人员扮演的角色和职位。
- 角色功能表：某个角色在系统中所具有的系统操作功能。
- 用户角色表：用户在使用系统时，所扮演的角色。

这些表通过用户信息表的 id 和其他扩展表相关联，每个角色对应具体的功能，至少具有基本权限，每个用户至少拥有一个角色，这样管理用户的权限功能很方便。

设置一个系统管理员用户，该用户具有所有的管理权限，而授权用户只有系统管理员，该用户可以对用户进行管理和用户权限进行管理。对用户的权限判断，首先读取用户角色表取得用户的角色数据，通过角色功能表获得用户所扮演的角色所具有的系统功能。系统由系统管理员进入进行功能的维护（添加、修改、删除）、角色的维护（添加、修改、删除）、用户的维护（添加、修改、删除）、角色权限的维护（添加、修改、删除）和用户角色的维护（添加、修改、删除）。

6.2　界面设计

注册及登录验证模块程序的界面分为用户注册界面和登录验证界面，注册成功后可以显示用户的注册信息，登录成功后进入下一步操作界面。

6.2.1　用户注册界面设计

用户注册界面的作用是引导新用户完成各项信息的填写，以及部分用户初始设置（如用户登录密码等），设计的核心是表单设计。表单将用户填写后的信息提交到服务端程序作相应处理，最终完成注册过程。

创建一个 HTML 格式文件，命名为 register.html，用来做用户注册页面。页面的设计代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>Registering form</title>
```

```
</head>
<body>
<form name="send" method="post" action="register.php"
    onSubmit="return Check()">
<table width="330" border="0" align="center" cellpadding="5"
    bgcolor="#eeeeee">
    <font color="#FF6699">*</font>
    为必填项
    <tr>
        <td width="40%">用户名：</td>
        <td><input name="username" type="text" id="username"> <font
            color="#FF6699">*</font></td>
    </tr>
    <tr>
        <td>密码：</td>
        <td><input name="password" type="password" id="password">
        <font color="#FF6699">*</font></td>
    </tr>
    <tr>
        <td>重复密码：</td>
        <td><input name="cpassword" type="password" id="cpassword">
        <font color="#FF6699">*</font></td>
    </tr>
    <tr>
        <td>Email:</td>
        <td><input name="email" type="text" id="email"></td>
    </tr>
    <tr>
        <td colspan="2" align="center"><input type="submit" name="Submit"
            value="提交"> <input type="reset" name="reset" value="重置">
        </td>
    </tr>
</table>
</form>
<script language="javascript">
function Check()// 验证表单数据有效性的函数
{
    if (document.send.username.value == "") {
        window.alert('请输入用户名!');
        return false;
    }
    if (document.send.username.value.length < 2) {
        window.alert('用户名长度必须大于 2!');
        return false;
    }
    if (document.send.password.value == "") {
        alert('请输入密码!');
        return false;
    }
    if (document.send.password.value.length < 6) {
        alert('密码长度必须大于 6!');
        return false;
    }
    if (document.send.password.value != document.send.cpassword.value) {
        alert('确认密码与密码不一致!');
        return false;
    }
    if (document.send.email.value == "") {
        alert('请输入 Email!');
```

```
        return false;
    }
    if (document.send.Email.value.indexOf("@") == -1) {
        alert('请输入有效的 Email 地址!');
        return false;
    }
    if (document.send.code.value == "") {
        alert('请输入验证码!');
        return false;
    }
    return true;
}
</script>
</body>
</html>
```

【代码解读】

其中，“charset=gb2312”这段代码的含义是，告诉浏览器本页使用 gb2312 编码（简体中文）来显示文本内容，即向浏览器声明文件所使用的字符集。在<form></form>标签的中间的部分就是 HTML 表单的内容。标签的 name 属性用以指出表单的名称，在客户端编程时引用表单内容。method 属性指出表单内的数据将以何种方式提交到服务器端，本例中使用的 POST 方式提交数据。action 属性表示浏览者的数据将被传送的位置，本例中的 action="register.php"，含义是将表单中的数据交给 register.php 文件来处理。

使用 IE 浏览加载 register.html 页面，如图 6.1 所示，其中*（红色）号为必填项。



图 6.1　用户注册界面

6.2.2　登录验证界面设计

用户登录界面的作用是核实用户身份、限制用户浏览部分内容。登录界面分为两个部分，即登录表单代码和对输入信息验证的 JavaScript 程序。

创建登录表单文件，它是扩展名为 html 的页面文件，代码如下：

```
<script language='javascript' src='check data.js'></script>
<script language="javascript">
<!--
function Check(){
    if(document.Login.username.value==""){
        alert('请输入你的用户名! ');
        return false;
    }
    if(document.Login.pwd.value==""){
        alert('请输入你的密码! ');
        return false;
    }
    if (document.Login.username.value.length<2)
    {
        window.alert('用户名长度必须大于 2!');
        return false;
    }
    if (document.Login.pwd.value.length<6)
```

```
    {
        alert('密码长度不足！');
        return false;
    }
}
-->
</script>
<body>
<form name="Login" method="post" action="login.php"
    onSubmit="return Check();">
<table border="0" cellpadding="8" width="350" align="center">
    <tr>
        <td colspan="2" align="center" class="alert">用户登录界面</td>
    </tr>
    <td>用户名：</td>
    <td><input name="username" type="text" id="username"
        class="textinput" /></td>
    </tr>
    <tr>
        <td>密码：</td>
        <td><input name="pwd" type="password" id="pwd" class="textinput"
        /></td>
    </tr>
    <tr>
        <td colspan="2" align="center"><input type="submit" class="btn"
            value="登录">   <input type="reset" class="btn"
            value="重置"></td>
    </tr>
</table>
</form>
</body>
```

说明： 因为篇幅的限制，基本 HTML 表单的<html></html>、<head></head>等元素如无特殊说明不再给出。

将以上代码保存为 login.html 文件，并在浏览器中运行，显示页面中的登录表单如图 6.2 所示。



图 6.2　用户登录页面

6.3　客户端数据有效性验证

在实际应用中，当新注册用户在填写注册信息时，如果没有按照给定规则填写或者输入了非法的字符，就会给系统造成问题。通常会影响数据的有效性，有时甚至危害系统安全（如 SQL 注入等）。因此需要对注册信息做必要地验证，数据验证的方式可以分为客户

端验证和服务器端验证。

在服务器端执行数据有效性验证有一个缺陷，即需要往返服务器来检查用户输入的数据。在浏览器中进行数据有效性验证的优点是，完全不需要向服务器发送任何内容。因为在客户端做数据验证，可以在将数据提交给服务器之前就排除很多输入错误，从而减少网络流量，减轻服务器的负担，加快验证的速度，从而提升系统的性能。

6.3.1　JavaScript 通用过滤函数

在使用 JavaScript 进行客户端验证时，有许多代码的功能是重复的（比如输入验证）。如果在每个页面都嵌入相同的脚本代码，不仅不利于维护，而且降低了代码的可读性。因此，将功能复用高的部分独立出来，这在 JavaScript 中可以通过将通用代码放到 JavaScript 文件中来实现，在需要使用时加载该 JavaScript 文件即可。

现在来看一个最常见的信息过滤函数，该函数的功能是去掉输入信息时加入的空格字符。创建一个 check_data.js 文件，文件内容如下：

```
function MyString(str) {
    this.str = str;
    this.getString = function() {
        return this.str;
    }
    //截去串头部的空格
    function ltrim(str) {
        var i = 0;
        while (str.charAt(i) == ' ') {
            ++i;
        }
        return str.substring(i, str.length);
    }
    //截去串尾部的空格
    function rtrim(str) {
        var i = str.length - 1;
        while (str.charAt(i) == ' ') {
            --i
        };
        return str.substring(0, i + 1);
    }
    //截去串头尾的空格
    function trim(str) {
        return ltrim(rtrim(str));
    }
    this.str = trim(this.str);     //调用成员函数
}
```

【代码解读】

上述代码中定义了 3 个方法，通过这 3 个方法分别去除字符串的头部、尾部和头尾存在的空格，完成后返回处理后的字符串。

现在写一个测试脚本测试这个过滤函数，建立一个测试脚本 check_test.html 文件，内容如下：

```
<html>
<head>
```

```
<meta http-equiv="content-type" content="text/html; charset=gb2312">
<title>JavaScript 多余空格过滤测试</title>
<script language='JavaScript' src='check data.js'></script>
<script language='JavaScript'>
    var objTest = new MyString('  测试 JavaScript 的过滤函数  ');
    var strTest = '|' + objTest.getString() + '|';
</script>
</head>
<body onLoad="JavaScript:alert(strTest);">
    JavaScript 多余空格过滤测试
</body>
</html>
```

使用 IE 浏览器加载该文件，预览效果如图 6.3 所示，通过图中可以看到程序正确地处理了输入的字符串。

6.3.2　表单数据的有效性验证

在数据被送往服务器端处理前，需要使用 JavaScript 对 HTML 表单中的这些输入数据进行有效性验证。使用 JavaScript 进行以下验证：

- 用户是否已填写表单中的必填项目。
- 用户输入的邮件地址是否合法。
- 用户是否已输入合法的日期。
- 用户是否在数据域（numeric field）中输入了文本。

图 6.3　多余空格验证函数

下面介绍实现客户端的表单验证的步骤，具体步骤如下。

（1）打开前面创建的 register.html 文件，找到文件中内容为：

```
<form name="send" method="post" action="register.php">
```

在该行添加触发事件，修改后代码如下：

```
<form name="send" method="post" action="register.php" onsubmit= "return
Check();">
```

【代码解读】

当 onsubmit 事件在提交表单时被触发。添加在代码的<form>标签中的 onsubmit 属性用来指定当 onsubmit 事件被触发时要执行的客户端代码“return Check();”，即执行 Check() 函数并返回该函数的返回值。当返回值为 true 时表单被正式提交，当返回值为 false 时表单的提交操作被终止（数据未提交到服务器端）。

（2）需要在 register.html 文件的</head>标签之前，添加自定义的 JavaScript 函数 Check()。该函数用于真正验证表单中输入数据的有效性，代码如下：

```
<script language="javascript">
    function Check() // 验证表单数据有效性的函数
    {
        if (document . send . username . value == "") {
            window . alert ( '请输入用户名!' );
            return false;
        }
```

```
        if (document . send . username . value . length < 2) {
            window . alert ( '用户名长度必须大于 2!' );
            return false;
        }
        if (document . send . password . value == "") {
            alert ( '请输入密码!' );
            return false;
        }
        if (document . send . password . value . length < 6) {
            alert ( '密码长度必须大于 6!' );
            return false;
        }
        if (document . send . password . value != document . send . cpassword .
        value) {
            alert ( '确认密码与密码不一致!' );
            return false;
        }
        if (document . send . Email . value == "") {
            alert ( '请输入 Email!' );
            return false;
        }
        if (document . send . Email . value . indexOf ( "@" ) == - 1) {
            alert ( '请输入有效的 Email 地址!' );
            return false;
        }
        if (document . send . code . value == "") {
            alert ( '请输入验证码!' );
            return false;
        }
        return true;
    }
</script>
```

【代码解读】

以上代码中，程序首先取得<form>表单中各输入域的值，然后判断各域的值是否为空，如果为空则弹出提示对话框并返回 false 终止提交。随后验证两次输入的密码是否一致，以及密码的长度是否为 6~30 个字符。最后，验证 Email 的格式是否正确，如果所有验证都通过，则在函数的最后返回 true，允许表单数据提交。

（3）在浏览器中运行 register.html，并在表单中输入测试数据，如图 6.4 所示。

其他几种输入不合法导致验证不能通过的情况，读者可以自行测试，检查客户端验证是否达到了预期的结果。



图 6.4　客户端数据有效性验证 Email 格式错误

6.4　服务端数据有效性验证

6.3 节对客户端提交的数据进行了客户端的验证，对表单输入信息的格式和多余空格

符号做了相应的过滤。但是客户端的验证有其局限性，验证范围也有限，因此一个完整的注册信息验证过程，还包括在服务器端对提交过来的数据进行有效性验证。服务数据有效性验证通常分为 3 个步骤，即正则表达式判断、用户名排重检测、SQL 注入验证。下面分别介绍这个 3 个步骤的具体内容。

6.4.1　正则表达式判断

正则表达式就是用某种模式去匹配一类字符串的一个公式。在 PHP 中，利用这些定制的公式可以更灵活地匹配、检验、替换和修改字符串。下面介绍的是由 PCRE（Perl Compatible Regular Expression）库提供的使用 preg_为前缀命名的函数。

PHP 中正则表达式操作函数使用频率较高的是 preg_match()、preg_replace()和 preg_split()这几个函数。在 PCRE 中，通常将模式表达式（即正则表达式）包含在两个“/”之间，如“/apple/”。下面具体介绍这几个常用的函数。

1．规则匹配preg_match

preg_match()用来完成字符串的规则匹配。当有符合的匹配时 preg_match()函数返回 1，否则返回 0。该函数的表达式如下：

```
int preg_match ( string pattern, string subject [, array matches [, int
flags]] )
```

其中第 3 个参数是必要参数，作用是把匹配的内容以数组的方式存储。在验证数据时会经常被使用，来看一个实际使用的代码片段：

```
<?php
$string = "helloworld";
if (preg_match ( '/hello/', $string )) {
    echo ‘得到正确匹配’;
}
?>
```

上面的例子将成功匹配，因为“helloworld”字符串中包含“hello”字符。

2．规则替换preg_replace

preg_replace()的作用是替换字符串中匹配到预先定义的正则表达式，其表达式如下：

```
mixed preg_replace ( mixed pattern, mixed replacement, mixed subject [, int
limit] )
```

其中，在 subject 中搜索 pattern 模式的匹配项，并替换为 replacement。如果指定了 limit，则仅替换 limit 个匹配，如果省略 limit 或者其值为–1，则所有的匹配项都会被替换。

下面的例子演示如何将 HTML 标签代码转换成文本形式，代码如下：

```
<?php
    //$document 应包含一个 HTML 文档
    //本例将去掉 HTML 标记，JavaScript 代码和空白字符。还会将一些通用的 HTML 实体
      转换成相应的文本
    $search = array ("'<script[^>]*?>.*?</script>'si", //去掉 JavaScript
```

```
    "'<[\/\!]*?[^<>]*?>'si",                              //去掉 HTML 标记
    "'([\r\n])[\s]+'",                                    //去掉空白字符
    "'&(quot|#34);'i",                                    //替换 HTML 实体
    "'&(amp|#38);'i",
    "'&(lt|#60);'i",
    "'&(gt|#62);'i",
    "'&(nbsp|#160);'i",
    "'&(iexcl|#161);'i",
    "'&(cent|#162);'i",
    "'&(pound|#163);'i",
    "'&(copy|#169);'i",
    "'&#(\d+);'e");
                                                          //作为 PHP 代码运行
    $replace = array ("",
    "",
    "\\1",
    "\"",
    "&",
    "<",
    ">",
    " ",
    chr(161),
    chr(162),
    chr(163),
    chr(169),
    "chr(\\1)");
    $text = preg_replace ($search, $replace, $document);
?>
```

这段代码可以对非法输入的 HTML 代码字符做转换，在实际开发中经常会被用到，读者注意在不同情况下过滤规则的设定。

3．规则分割preg_split

preg_split()可以将整段字符串按匹配到的正则表达式，分割成一两个或更多字符的多段。函数表达式如下：

```
array preg_split ( string pattern, string subject [, int limit [, int flags]] )
```

该函数返回一个数组，包含 subject 中沿着与 pattern 匹配的边界所分割的子串。如果指定了 limit，则最多返回 limit 个子串。如果 limit 是–1，则意味着没有限制，可以用来继续指定可选参数 flags，如获取标签等，无论是用空格还是逗号分隔的。示例代码如下：

```
<?php
    $keywords = preg_split('/[,]/', 'my,tags,unevenly,spaced');
    print($keywords);
?>
```

正则表达式包含的内容十分广泛，其中涉及的规则也很复杂，读者可以对照正则表达式的规则多做练习，在实践中完成更复杂应用。

了解了以上正则表达式的基础知识，现在可以编写代码来实现服务器端数据有效性的验证。创建文件名为 register.php 的程序，用来验证客户端提交过来的数据，文件代码如下：

```
<?php
    //trim()函数可以截去头尾的空白字符
    $username = trim($_POST['username']);
```

```
    $password = $_POST['password'];
    $cpassword = $_POST['cpassword'];
    $email = trim($_POST['email']);
    if (empty($username) || empty($email)||
                                //数据验证，empty()函数判断变量内容是否为空
    empty($password) || $cpassword  !=
    $password)
    {
        echo '数据输入不完整';
        exit;
    }
    else
    {
        //密码长度判断
        if (strlen($password) < 6 ||
        strlen($password) > 30)
        {
            echo '密码必须在 6 到 30 个字符之间';
            exit;
        }
        //与客户端验证 Email 时相同的正则表达式
        $pattern ="/^\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$/";
        if (!preg_match($pattern, $email))
        {
            echo 'Email 格式不合法!';
            exit;
        }
    }
?>
```

【代码解读】

以上代码综合使用本节介绍的关于正则表达式的函数，验证程序首先对提交的字符串做基础的验证（如字符串完整性、密码长度等），然后根据设定的正则规则使用 preg_match() 函数验证字符串的有效性。

其中以下两段代码请读者自行替换连接的数据库、用户名、密码口令等。

```
$db  =  mysql_connect('mysql_host',  'mysql_user',  'mysql_password')  or
die('Could not connect: ' . mysql_error());
mysql_select_db('my_database') or die('Could not select database');
```

注意：正则表达式的功能虽然强大，但是如果使用普通的字符串操作函数也能完成同样的功能就不必使用正则表达式。过度地使用正则表达式会降低代码的执行效率，在实际应用中需要灵活掌握。

6.4.2　用户名排重检测

在新用户注册时，有时会出现想要注册的用户名已经被注册的情况，所以在注册前检查用户名是否已存在是十分必要的。同样，在一个数据库中的用户名也应当是唯一的，因此需要检查 user 表中的 username 字段，以判断用户名是否已经存在。打开 register.php 文件，添加用户名是否存在的检测。改造后的 register.php 文件代码如下：

```
<?php
    //trim()函数可以截去头尾的空白字符
```

```
$username = trim($_POST['username']);
$password = $_POST['password'];
$cpassword = $_POST['cpassword'];
$email = trim($_POST['email']);
//数据验证，empty()函数判断变量内容是否为空
if (empty($username) || empty($email)||
empty($password) || $cpassword  !=
$password)
{
    echo '数据输入不完整';
    exit;
}
else
{
    //密码长度判断
    if (strlen($password) < 6 ||
    strlen($password) > 30)
    {
        echo '密码必须在 6 到 30 个字符之间';
        exit;
    }
    //与客户端验证 Email 时相同的正则表达式
    $pattern ="/^\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*$/";
    if (!preg_match($pattern, $email))
    {
        echo 'Email 格式不合法!';
        exit;
    }
    //创建数据库连接
    $db = mysql_connect('localhost','root', '198251') or die('Could not
    connect: ' . mysql_error());
    //echo 'Connected successfully';
    mysql_select_db('register') or die('Could not select database');
    //查询数据库，看填写的用户名是否已经存在
    $sql = "SELECT * FROM 'user' WHERE 'username' = '".$username."'";
    $result = mysql_query($sql);
    if ($result && mysql_num_rows($result)> 0)
    {
        echo "<font color='red'size='5'>该用户名已被注册，请换一个重试!
        </font><br>\n";
        echo $username."<br>\n";
        echo $password."<br>\n";
        echo $cpassword."<br>\n";
        echo $email."<br>\n";
    }
    else
    {
        //将用户信息插入数据库的 user 表
        $sql = "INSERT INTO user (username,pwd,email) VALUES";
        $sql  .= "('$username', '$password','$email')";
        //echo $sql;
        $result = mysql_query($sql);
        if (!$result)
        {
            mysql_free_result($result);     //释放结果集
            mysql_close($db);               //关闭连接
            echo '数据记录插入失败!';
            exit;
        }
```

```
            echo "<font color='red' size='5'>恭喜您注册成功!</font><br>\n";
        }
        mysql_close($db);                       //关闭数据库连接
    }
?>
```

【代码解读】

在上面的代码中，通过下面的 SQL 语句：

```
$sql = "SELECT * FROM 'user' WHERE 'username' = '".$username."'";
```

通过获得的查询结果是否大于 0，可知用户名是否已经存在。如果存在则仅给出提示，如果不存在则将用户信息记录到数据库中。现在来看再次注册同一个用户的情况。在浏览器中运行 register.php 文件，并在页面的表单中输入重复的用户名，并单击“提交”按钮，结果如图 6.5 所示。

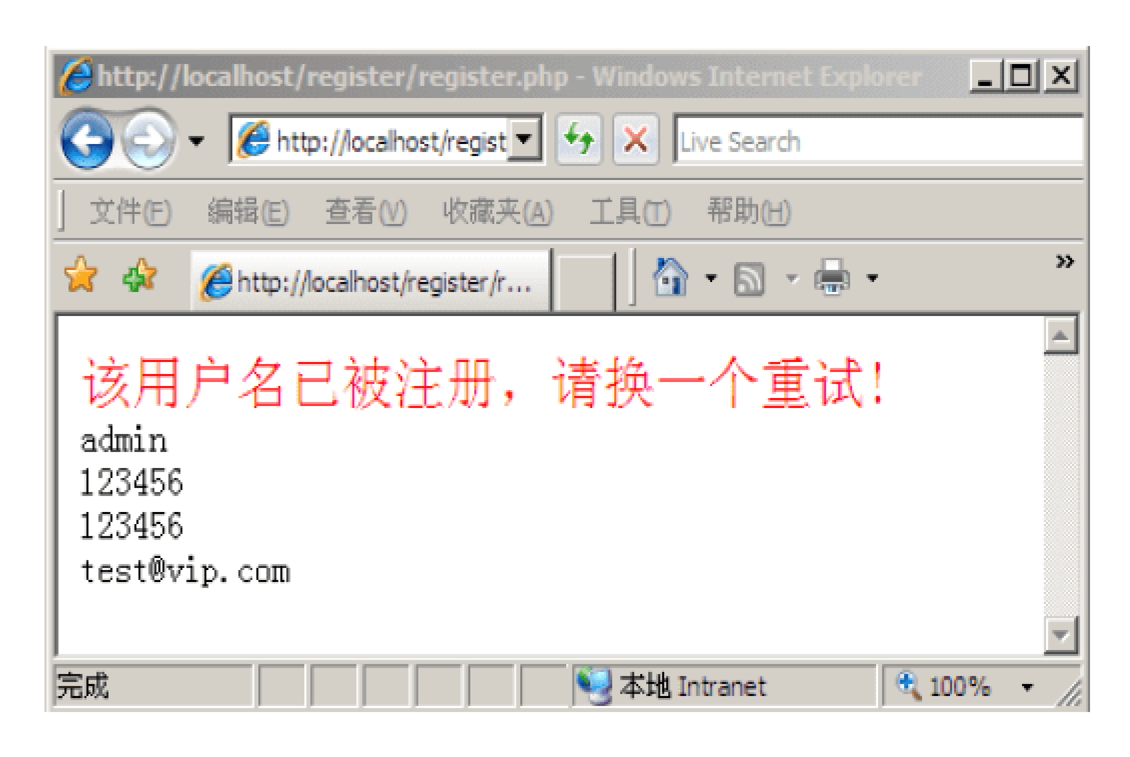


图 6.5　提示用户名已存在页面

6.4.3　SQL 注入验证

前面对客户端提交的数据进行了有效性验证，主要内容是截去头尾空格，然后判断是否为空、密码长度是否在有效范围、Email 的格式是否有效，以及用户名是否重复等，这些都是常规的数据验证。SQL 注入漏洞则是一个要特别注意的问题。客户端提交到服务器处理的数据往往被用于构造 SQL 语句，使用这些构造出来的 SQL 语句访问数据库时会发生 SQL 注入攻击。

说明：SQL 注入是客户端提交的数据构成了非法访问数据库的 SQL 语句。

为防止 SQL 注入漏洞，需要注意以下两点。

- 遵循最小权限的原则，即赋予连接数据库的用户尽可能小（仅能执行预期的操作）的权限，严禁使用特权用户操作（如 root）；
- 尽可能地过滤由客户端提交的可疑的非法数据。假设在服务器端将执行如下 SQL 语句：

```
$sql = "DELETE FROM user WHERE username='$username'";
```

【代码解读】

这条 SQL 语句的本意是根据客户端的要求删除特定用户，但是如果一个不怀好意的访

问者知道服务器端的数据库的用户信息表的名称，而且提交了一个用户名为“1' OR 1 --”的数据，即“$username = "1' OR 1 --"”，此时将上面这条 SQL 语句中的$username 替换成提交上来的实际值后成为：

```
$sql = "DELETE FROM t_user WHERE username='1' OR 1 -- '";
```

由于删除条件中的 OR 1，结果表 user 中的记录将被全部删除。

要防范 SQL 注入漏洞，实际上只要屏蔽一些 SQL 命令及关键字即可。在进行服务器端的数据有效性验证之前，调用以下函数即可防止 SQL 注入漏洞，程序代码如下：

```
<?php
    function checkIllegalWord ()
    {
        //定义不允许提交的 SQL 命令及关键字
        $words = array();
        $words[] = " add ";
        $words[] = " count ";
        $words[] = " create ";
        $words[] = " delete ";
        $words[] = " drop ";
        $words[] = " from ";
        $words[] = " grant ";
        $words[] = " insert ";
        $words[] = " select ";
        $words[] = " truncate ";
        $words[] = " update ";
        $words[] = " use ";
        $words[] = "-- ";
        //判断提交的数据中是否存在以上关键字，$_REQUEST 中含有所有提交数据
        foreach($_REQUEST as $strGot)
        {
            $strGot = strtolower($strGot); //转为小写
            foreach($words as $word)
            {
                if (strstr($strGot, $word))
                {
                    echo "您输入的内容含有非法字符！";
                    exit;                   //退出运行
                }
            }
        }
        //foreach
    }
    checkIllegalWord();         //在本文件被包含时即自动调用
?>
```

输入以上代码，并将其保存成 function.php 文件，在需要验证的页面文件中只需要简单地引用该文件即可。引用的语法如下：

```
require_once('function.php);
```

require_once()函数可以保证文件仅能被引用一次，如果引用出错会导致一个致命级的错误，终止下面代码的运行，这样可以在逻辑层面保证程序的安全性。

6.5　已注册用户登录

当新用户注册成为会员之后，就会获得网站相应的会员权限，这样就可以进行后续的操作。通过前面的学习，读者应该了解了在客户端和服务端进行数据有效性验证的具体方法。本节将使用前面创建的公共函数来实现完整的登录模块程序。

6.5.1　客户端用户登录验证

创建客户端验证程序，前面在登录界面的设计中已经做了初步的客户端验证，完整的代码如下：

```
<script language="JavaScript">
function Check(){
    if(document.Login.username.value==""){
        alert('请输入你的用户名！');
        return false;
    }
    if(document.Login.pwd.value==""){
        alert('请输入你的密码！');
        return false;
    }
}
</script>
```

【代码解读】

以上代码将保证输入的用户名和密码不为空，这显然是不够的。

接下来继续扩展该 JavaScript 函数，细心的读者一定还记得在处理注册页面的时候做过类似的验证，根据上面的代码继续完成改造。打开 login.html 文件添加完整的客户端验证代码，改造后的完整代码如下：

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>User Login</title>
<script language='JavaScript' src='check data.js'></script>
<script language="JavaScript">
function Check(){
    if(document.Login.username.value==""){
        alert('请输入你的用户名！');
        return false;
    }
    if(document.Login.pwd.value==""){
        alert('请输入你的密码！');
        return false;
    }
    if (document.Login.username.value.length<2)
    {
        window.alert('用户名长度必须大于 2!');

        return false;
    }
```

```
    if (document.Login.pwd.value.length<6)
    {
        alert('密码长度不足！');

        return false;
    }
}
</script>
</head>
<body>
<form  name="Login"  method="post"  action="login.php"  onSubmit="return
Check();">
<table border="0" cellpadding="8" width="350" align="center">
    <tr>
        <td colspan="2" align="center" class="alert">用户登录界面</td>
    </tr>
      <td>用户名：</td>
      <td><input name="username" type="text" id="username" class=
      "textinput" /></td>
    </tr>
<tr><td>密码：</td>
    <td><input name="pwd" type="password" id="pwd" class="textinput" />
    </td>
</tr>
<tr><td colspan="2" align="center">
    <input type="submit" class="btn" value="登录">  
    <input type="reset" class="btn" value="重置">
    </td>
</tr>
</table>
</form>
</body>
```

打开浏览器测试，当没有输入密码时无法登录并提示消息，测试结果如图 6.6 所示。从图中可以看到，客户端程序有效的验证了输入的字符串。

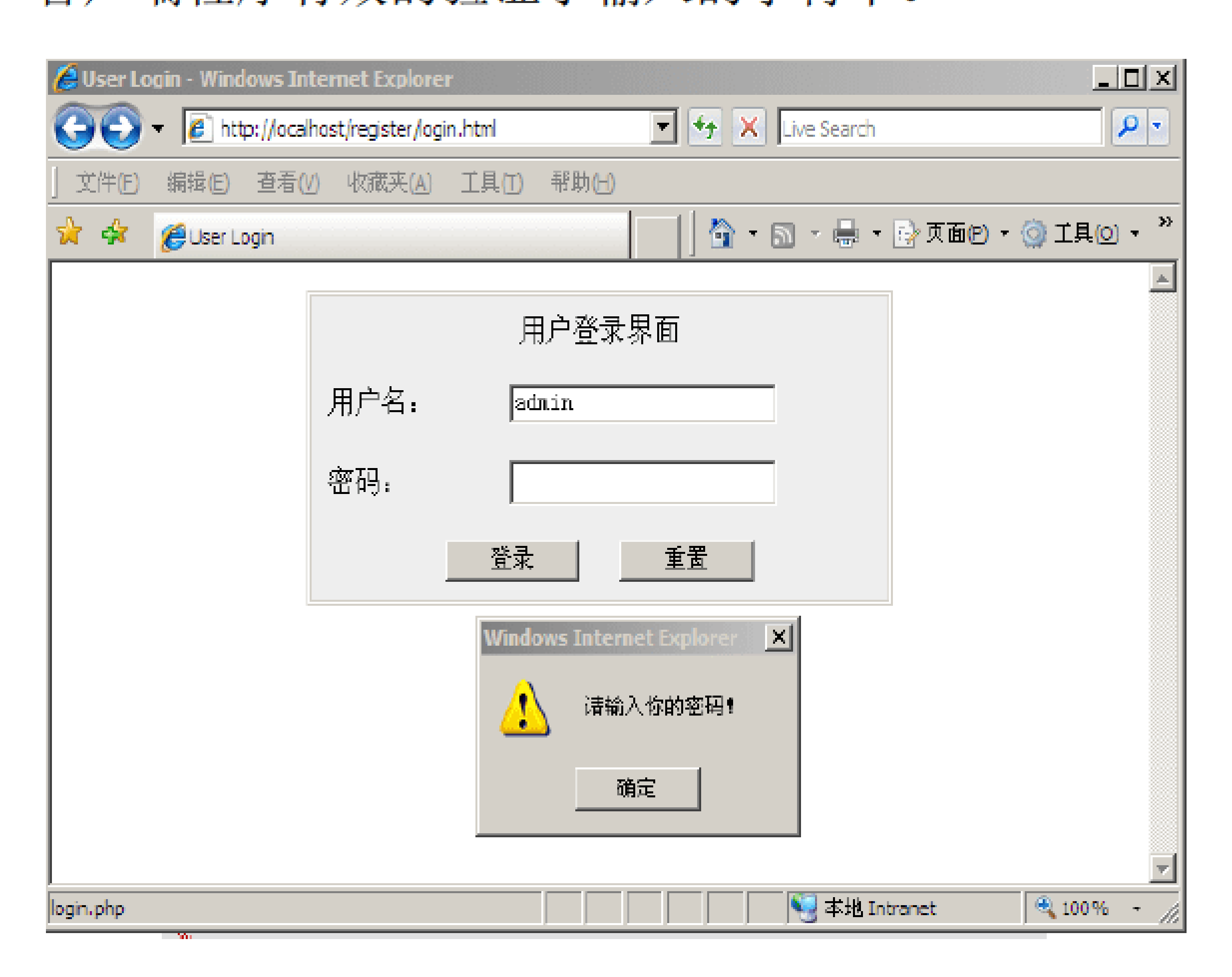


图 6.6　登录界面客户端数据验证

6.5.2　服务端用户登录状态验证

用户在登录表单中输入登录信息之后，数据被提交到 login.php 进行处理，创建一个名

为 login.php 的文件，代码如下：

```
<?php
/*文件功能：
 *加载防止 SQL 注入漏洞检查的代码
 */
require_once('function.php');
$username = trim($ POST['username']);  //取得客户端提交的用户名
$password = trim($_POST['pwd']);       //取得客户端提交的密码
//设置一个错误消息变量，以便判断是否有错，在客户端显示错误消息。 其初值为空
$errmsg = '';
if (!empty($username))
{
    //用户填写了数据才执行数据库操作
    //数据验证，empty()函数判断变量内容是否为空
    if (empty($username))
    {
        $errmsg = '数据输入不完整';
    }
    if (empty($errmsg))
    {
        //$errmsg 为空说明前面的验证通过
        //创建数据库连接
        $db = mysql connect('localhost','root', 'pwd') or die('Could not
        connect: ' . mysql error());
        //echo 'Connected successfully';
        mysql select db('register') or die('Could not select database');
        //检查数据库连接
        if (mysqli connect errno())
        {
            $errmsg = "数据库连接失败!\n";
        }
        else
        {
            //查询数据库，看用户名及密码是否正确
            $sql = "SELECT * FROM user WHERE username='$username' AND
            pwd='$password'";
            $result = mysql query($sql);
            if ($result && mysql num rows($result)> 0)
            {
                $errmsg = "登录成功!";        //提示消息
                else
                {
                    $errmsg ="用户名或密码不正确，登录失败!";
                }
                mysql free result($result);//释放资源
                mysql_close($db);          //关闭连接
            }
        }
    }
    echo "<font color='red'size='5'>用户: ".$username."".$errmsg."
    </font><br>\n";
?>
```

【代码解读】

以上代码首先对客户端提交过来的数据值进行基础验证（如数据完整性的验证），然后

对客户端提交过来的数据与数据库中的记录对比，如果正确，返回结果，否则抛出报错信息。

6.5.3　标记登录状态

对用户而言，之所以要注册成为会员是为了享受会员的服务，因为不同的用户拥有的权限也是不同的。因此在登录成功后需要注册一个全局变量，该变量用来记录用户的身份和权限信息，这个全局变量就是程序判断用户身份及权限的依据。

HTTP 协议是“一次性单向”协议。服务端不可以主动连接客户端，只能被动等待并答复客户端请求。HTTP 协议本身并不能支持服务端保存客户端的状态信息。于是，B/S 结构的应用中引入了 Session 的概念，用来保存客户端的状态信息。Session 的典型应用是存放用户的 Login 信息，如用户名、密码、权限角色等信息，应用程序（如 Email 服务、网上银行等系统）根据这些信息进行身份验证和权限验证。

在 PHP 中，使用 Session 非常简单。PHP 提供了一个自动全局变量$_SESSION 用于处理 Session。但是要注意，如果在 PHP 的配置文件中没有设置自动启动 Session 的话，在使用 Session 之前一定要调用 session_start()函数创建 Session 变量。标记登录状态的操作步骤如下。

（1）首先打开 login.php 文件，该程序需要完成的功能是在登录成功后创建记录状态的 Session 变量，该文件的代码如下：

```
<?php
/*文件功能：
 *加载防止 SQL 注入漏洞检查的代码
 */
require_once('function.php');
$username = trim($_POST['username']);  //取得客户端提交的用户名
$password = trim($_POST['pwd']);       //取得客户端提交的密码
//设置一个错误消息变量，以便判断是否有错，在客户端显示错误消息。 其初值为空
$errmsg = '';
if (!empty($username))
{
    //用户填写了数据才执行数据库操作
    //数据验证，empty()函数判断变量内容是否为空
    if (empty($username))
    {
        $errmsg = '数据输入不完整';
    }
    if (empty($errmsg))
    {
        //$errmsg 为空说明前面的验证通过
        //创建数据库连接
        $db = mysql_connect('localhost','root', '198251') or die('Could not
        connect: ' . mysql_error());
        //echo 'Connected successfully';
        mysql_select_db('register') or die('Could not select database');
        //检查数据库连接
        if (mysqli_connect_errno())
        {
            $errmsg = "数据库连接失败!\n";
```

```
        }
        else
        {
            //查询数据库，看用户名及密码是否正确
            $sql = "SELECT * FROM user WHERE username='$username' AND
            pwd='$password'";
            $result = mysql_query($sql);
            if ($result && mysql_num_rows($result)> 0)
            {
                $errmsg = "登录成功!";               //提示消息
                session_start();                    //注册 session 变量
                $_SESSION['login'] = 'true';        //标识登录状态 true 为已经登录
                $_SESSION['user'] = $username;      //记录该用户信息
                else
                {
                    $errmsg ="用户名或密码不正确，登录失败!";
                }
                mysql_free_result($result);         //释放资源
                mysql_close($db);                   //关闭连接
            }
        }
    }
    echo "<font color='red'size='5'>用户: ".$username."".$errmsg."
    </font><br>\n";
?>
```

（2）当已注册用户登录成功后，程序会在服务端创建记录状态和身份信息的全局变量，并将用户名保存到 Session 中。

判断用户是否已经登录的代码十分简单，代码如下：

```
<?php
    session_start();
    if (empty($_SESSION['login'])) {
        echo "您还没有登录，不能访问当前页面！";
        exit;
    }
?>
```

【代码解读】

通过判断自动全局变量$_SESSION['login']是否为空，就可以判断该用户是否已经登录。如果用户没有登录，提示其无法访问当前页面，并终止程序的运行（或者使用一条重定向语句将页面导向登录页）。

（3）下面创建一个名字为 test.php 的测试脚本：

```
<?php
    require ('is_login.php');        //登录状态测试脚本
    echo '如何您还没有登录您将看不到这条消息；）';
    unset( $_SESSION['login']);      //调试登录状态
?>
```

执行以上代码，当已经登录过就会看到“如果您还没有登录您将看不到这条消息”，否则就看到没有登录的提示信息，程序也会终止运行。

6.6　小　　结

本章主要是以用户注册和登录验证为例，介绍常见的通用功能程序并封装，封装后可以以接口的方式在各个平台中使用。在具体技术层面，笔者主要介绍了数据有效性验证的基本原理和实现方法。

本章首先介绍用户注册及登录验证的基本需求。第 6.1 节介绍了用户信息表，它是用户注册及验证的核心。第 6.2 节介绍了用户注册界面和登录验证界面的设计。第 6.3 节介绍了注册信息客户端数据验证，其中涉及两种验证方法，即 JavaScript 通用过滤函数和表单数据有效性验证。第 6.4 节介绍了数据验证的另一种模式（服务端数据的有效性验证），具体内容包含 3 个部分，即正则表达式判断、检查用户名是否已存在和 SQL 注入验证。最后将以上内容在第 6.5 节做了整合应用。

读者在学习本章时，需要了解新用户注册、客户端数据验证、服务端数据验证、PHP 中正则表达式的实现、已注册用户登录、登录状态及客户信息保存等知识点。

第 7 章　上传与下载模块（PHP 防盗链+ActiveXObject 控件）

上传和下载是互联网上最常见的服务，该服务不仅方便了用户的使用，同时也丰富了网站资源。通过文件的上传和下载功能，可以使资源更好地共享，使网站成为真正的开放平台。上传部分和下载部分的原理有很大的不同，本章将从上传和下载的原理讲起，由浅入深地展开，最终将上传功能与下载功能整合成一个完整的应用平台。

本章主要涉及的知识点有：

- PHP 实现文件上传的原理。
- PHP 防盗链的程序实现原理。
- JavaScript 文件尺寸判断核心 ActiveXObject()控件。
- 在 PHP 中使用 header()函数向浏览器发送原始 HTTP 标头。

7.1　上传与下载的实现原理

在 B/S 结构的程序开发中，文件上传的原理都是相同的。本节将讲解基于 PHP 的文件上传原理并实现编码。文件下载部分原理并不复杂，但是如果直接开放链接给用户下载就会造成很多的问题，所以提供基于防盗链的下载链接十分必要，通过防盗链可以更好地保护网站资源。

7.1.1　PHP 文件上传的原理及实现

在 PHP 中，文件上传功能是使用 PHP 提供的文件函数来实现的。下面通过例子使读者直观地了解上传的原理以及程序编码的实现。

1．客户端上传

创建一个文件 upload.html，这是一个最基本的上传的表单代码，代码如下：

```
<html xmlns="undefined">
<head>
<title>文件上传页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta name="description" content="文件上传页面" />
<style type="text/css">
body,td{font-family:tahoma,verdana,arial;font-size:11px;line-height:
15px;background-color:white;color:#666666;
```

```
strong{font-size:12px;}
a:link{color:#0066CC;}
a:hover{color:#FF6600;}
a:visited{color:#003366;}
a:active{color:#9DCC00;}
a{TEXT-DECORATION:none}
td.irows{height:20px;background:url("index.php?i=dots") repeat-x bottom}
</style>
</head>
<body bgcolor="#FFFFFF">
<center>
  <form enctype="multipart/form-data" method="post" name="upform"
  action="">
    <table border="1" width="55%" id="table1" cellspacing=0>
      <tr>
        <td colspan="2">
            <p align="center">上传文件:
        </td>
      </tr>
      <tr>
        <td width="10%"></td>
        <td width="71%">
            <input type="hidden" name="max file size" value="100000">
            <input name="userfile" type="file">
            <input type="submit" value="上传文件">
        </td>
      </tr>
    </table>
 </form>
</center>
</body>
</html>
```

在浏览器中加载 upload.html 文件，效果如图 7.1 所示。

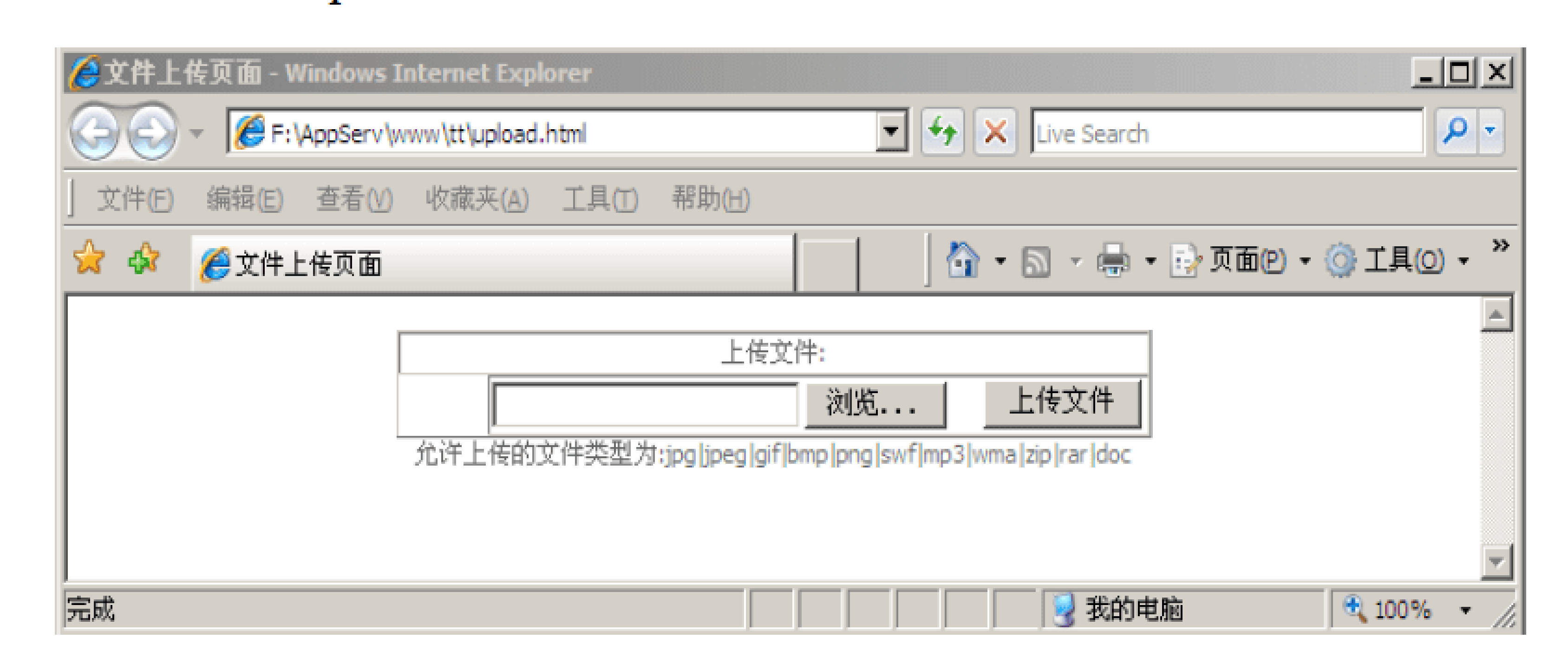


图 7.1　文件上传表单预览

要实现文件的上传，需要在表单标签中设置以下选项，以确保匿名上传文件的正确编码。

```
enctype="multipart/form-data"
```

【代码解读】

表单选项 MAX_FILE_SIZE 的隐藏值域，通过设置 Value 值的大小可以限制上传文件的尺寸。当然，MAX_FILE_SIZE 的值相对于浏览器只是一个参考值，实际上它可以被轻易地绕过。实际应用中是通过在 PHP 配置文件中设置上传文件最大值，来做上传文件的限

制的。在表单中加上 MAX_FILE_SIZE，可以及时发现上传文件尺寸过大的问题。

2. 服务器端上传

创建服务器端上传处理文件 upload.php，代码如下：

```
<?php
    $file = &$HTTP_POST_FILES ['userfile'];     //接收表单信息
    $dest_dir = 'up';                           //设定上传目录
    $dest = $dest dir . time () . $file ['name'];
                                        //设置文件名为时间加文件名，这样可有效避免重复
    if (! $file) {
        echo "<font color='red'>移动文件出错！</a>";
        exit ();
    } else {
        $r = move_uploaded_file ( $file ['tmp_name'], $dest );
        //chmod($dest, 0755);  //设定上传的文件的属性
        echo "<font color='red'>恭喜文件" . $dest . "上传成功！</a>";
    }
?>
```

【代码解读】

当文件上传成功后，系统会提示文件上传成功的信息。上面例子中$_FILES['userfile']数组的内容含义如下所示。

- $_FILES['userfile']['name']：客户端机器文件的原名称。
- $_FILES['userfile']['type']：文件的 MIME 类型，例如“image/gif”。
- $_FILES['userfile']['size']：已上传文件的大小，单位为 B。
- $_FILES['userfile']['tmp_name']：文件被上传后在服务端储存的临时文件名。
- $_FILES['userfile']['error']：该文件上传相关的错误代码。

其中，错误代码的含义如下所示。

- 值 0：没有错误发生，文件上传成功。
- 值 1：上传的文件超过了 php.ini 中 upload_max_filesize 选项限制的值。
- 值 2：上传文件的大小超过了 HTML 表单中 MAX_FILE_SIZE 选项指定的值。
- 值 3：文件只有部分被上传。
- 值 4：没有文件被上传。

7.1.2　PHP 文件下载的原理及实现

通常文件下载过程是十分简单的，建立一个链接指向到目标文件就可以了。例如下面的链接：

```
<a href=http://www.xxx.com/xxx.rar>点击下载文件</a>
```

但是，实际情况可能会稍复杂。比如需要用户填写完整注册信息后才可以下载该文件，这时最先想到的是使用 Redirect 的方式。下面介绍两种方式。

（1）用 Redirect 方式。先检查表格是否已经填写完毕和完整，然后将链接指到该文件，这样用户就可以下载。请看下面的示例代码：

```
<?php
/*文件功能：检查变量 form 是否完整*/
if($form){
//重新定向浏览器指向
Header("Location: http:// http://www.xxx.com/xxx.rar");
exit;
}
?>
```

（2）根据下载文件的序号来查找，链接的形式如下：

```
<a href="http://www.xxx.com/download.php?id=123456">点击下载文件</a>
```

上面的链接使用 ID 方式接收要下载文件的编号，然后再用 Redirect 的方式连接到真实的文件链接。

以上这两种方法虽然实现了文件的下载功能，但是缺点是直接暴露了文件所属的路径，而且没有防盗链的功能，所以上面的方式是简单直接但存在安全隐患的文件下载方式。在 PHP 中，通常是利用 header()函数和 fread()函数来实现安全的文件下载。

例如，需要下载的是一个文件名为 xxx.rar 的文件，首先创建文件是 download.php 的 PHP 文件。通过前面的例子很容易通过文件的 ID 号从数据库中得到待下载文件的真实位置，在获得文件的真实存储位置后，可以通过 header()函数的 location 参数直接重定向到这个文件。但是这样仍然是不安全的，因为某些下载软件还是可以通过重定向分析获得该文件的位置信息。因此需要用另外一种方法，就是 PHP 的文件处理 API 函数。它是通过 fread()函数把文件直接输出到浏览器提示用户下载，这样所有的处理都是在服务器端完成的，因此用户就无法获得文件具体存储位置信息的，示例代码如下：

```
<?
$file name = "xxx.rar";     //下载文件名
$file_dir = "./up/";        //下载文件存放目录
//检查文件是否存在
if (! file exists ( $file dir . $file name )) {
    echo "文件找不到";
    exit ();
} else {
    //打开文件
    $file = fopen ( $file dir . $file name, "r" );
    //输入文件标签
    Header ( "Content-type: application/octet-stream" );
    Header ( "Accept-Ranges: bytes" );
    Header ( "Accept-Length: " . filesize ( $file dir . $file name ) );
    Header ( "Content-Disposition: attachment; filename=" . $file_name );
    //输出文件内容
    //读取文件内容并直接输出到浏览器
    echo fread ( $file, filesize ( $file dir . $file name ) );
    fclose ( $file );
    exit ();
}
?>
```

【代码解读】

上述代码中，程序发送 Header 信息是用来告诉 Apache 和浏览器下载文件的相关信息的。content-type 的含义代表文件 MIME 类型是文件流格式。如果在 Apache 配置里面把文

件的 MIME 类型设为 application/octet-stream（如 add application/octet-stream .xxx.rar），那么浏览器（客户端）就会知道，这是一个文件流格式的文件并提示用户下载。Accept-Ranges 是一个响应头标，它允许服务器指明将在给定的偏移和长度处，为资源组成部分的接受请求，该头标的值被理解为请求范围的度量单位。Content-Length 是指定包含于请求或响应中数据的字节长度，例如，Content-Length:382。Content-Disposition:attachment 是用来告诉浏览器，文件是可以当做附件被下载，下载后的文件名称为$file_name 该变量的值。

运行 download.php 文件，效果如图 7.2 所示。从图中可以看到文件按照预想的方式被提示下载，单击“保存”按钮将文件保存在本地。

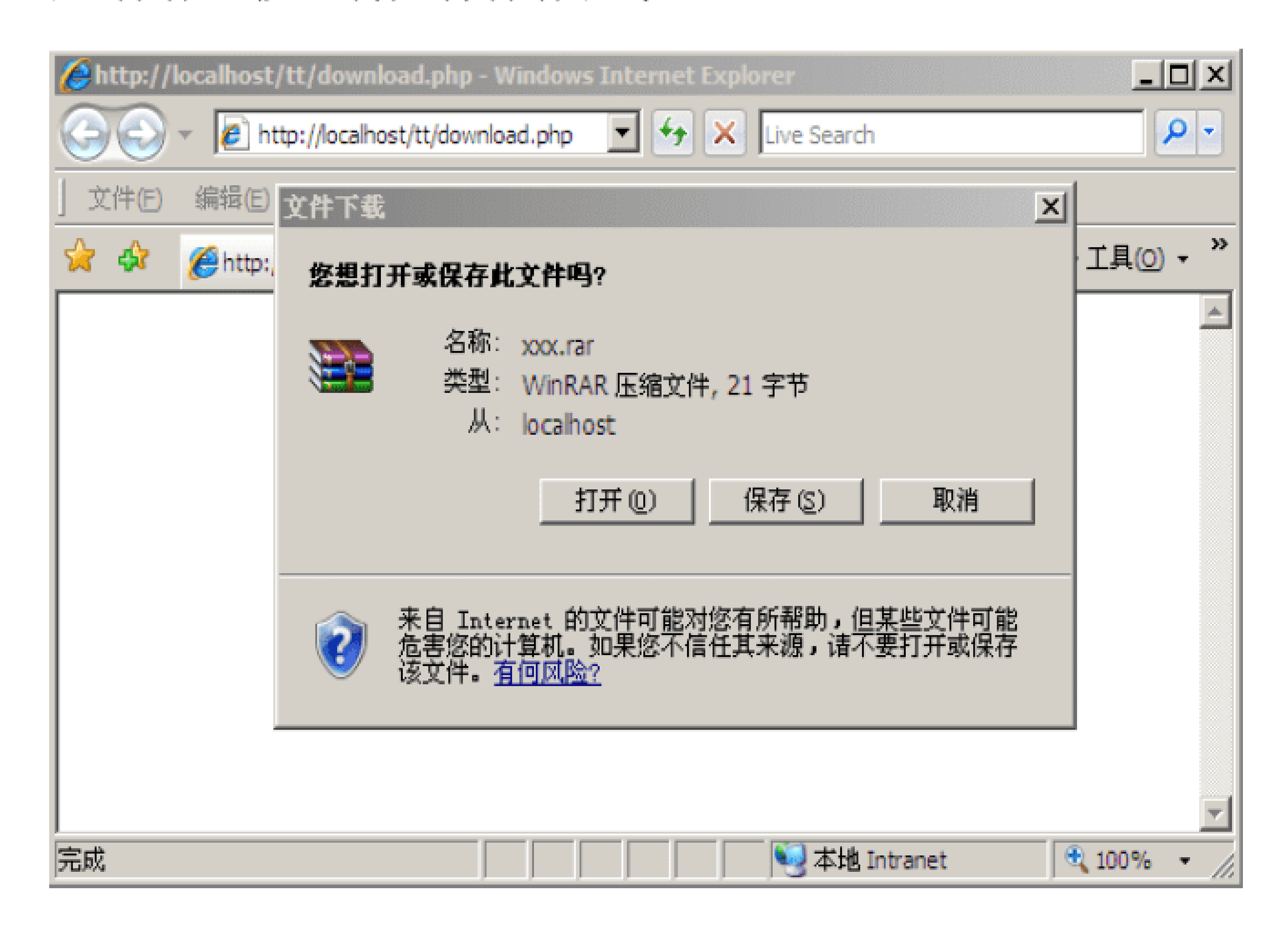


图 7.2　PHP 文件安全下载

7.2　文件尺寸限制及异常处理

在实际的应用中，文件管理功能通常需要对用户上传的文件尺寸和类型做限制，这是上传下载模块的基础功能。对于上传文件尺寸的限制可以在客户端依靠 JavaScript 程序来实现，但同时也需要对 WebServer 的配置做必要的修改，下面分别介绍这两种方式。

7.2.1　JavaScript 文件尺寸验证

打开前面创建的文件上传界面文件 upload.html，添加文件尺寸判断函数。修改后的页面代码如下：

```
<html xmlns="undefined">
<head>
<title>文件上传页面</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta name="description" content="文件上传页面" />
<script language="JavaScript" type="text/JavaScript">
                function getFileSize(filename) {
                    //var filename = document.all('fileup').value;
                                                            //获得上传文件的物理路径
```

```
                    if (filename == '') {
                        alert("你还没有浏览要上传的文件");
                        return false;
                    }
                    try {
                        var fso, f, fname, fsize;
                var flength = 4;
                            //设置上传的文件最大值（单位：kb），超过此值则不上传
                        fso = new ActiveXObject("Scripting.FileSystem-
                        Object");
                        f = fso.GetFile(filename); //文件的物理路径
                        fname = fso.GetFileName(filename);
                                            //文件名（包括扩展名）
                        fsize = f.Size;         //文件大小（bit）
                        fsize = fsize / 1024;
                        if (fsize > flength) {
        alert("上传的文件到小为：" + fsize + "kb,\n 超过最大限度" + flength + "kb,
        不允许上传 ");
                            return false;
                        } else {
                            alert("允许上传，文件大小为：" + fsize + "kb");
                        }
                    } catch(e) {
                        alert(e);
                        return false;
                    }
                    return true;
                }
            </script>
</head>
<body bgcolor="#FFFFFF">
<center>
<form enctype="multipart/form-data" action="upload.php" method="post">
<table border="1" width="55%" id="table1" cellspacing=0>
    <tr>
        <td colspan="2">
        <p align="center">上传文件:
        </td>
    </tr>
    <tr>
        <td width="10%"></td>
        <td width="71%"><input type="hidden" name="max file size"
            value="100000"> <input name="userfile" type="file"> <input
            type="submit" value="上传文件"
            onclick="getFileSize(document.all('userfile').value)"></td>
    </tr>
</table>
</form>
</center>
</body>
</html>
```

使用上面的程序上传一个大尺寸的文件，通过测试程序是否正确来判断文件尺寸。上传后浏览器弹出提示文件尺寸过大的信息，验证文件尺寸的检测有效。

注意：上面的代码做文件尺寸检测是依赖于 ActiveXObject("Scripting.FileSystemObject"); 这个是 IE 控件完成的，一些情况下需要降低 IE 的安全级别才能正常运行。

7.2.2　PHP 上传文件涉及的参数

PHP 默认的文件上传尺寸限制最大是 2M，如果需要上传超过该尺寸大小的文件，需要调整 PHP 的配置文件中的部分参数。下面介绍 PHP 文件上传中涉及需要调节的参数及含义。

- file_uploads：是否允许通过 HTTP 上传文件的开关，默认为 ON，即开通。
- upload_tmp_dir：用来指明 PHP 上传文件放置的临时目录，待上传的文件需要保证服务器没有关闭临时文件并对文件夹有“写”权限，如果未指定则 PHP 使用系统默认值。
- upload_max_filesize：允许上传文件大小的最大值，默认为 2M。
- post_max_size：设置在 POST 方法中，进行一次表单提交中 PHP 所能够接收的最大数据量。如果希望使用 PHP 文件上传功能，则需要将此值改为比 upload_max_filesize 数值大。
- max_input_time：以秒为单位对通过 POST、GET 及 PUT 方式接收数据的时间进行限制。如果应用程序所运行环境处在低速链路上，则需要增加此值以适应接收数据所需的更多时间。
- memory_limit：为了避免正在运行的脚本大量使用系统可用内存（PHP 允许定义内存使用限额）。通过 memory_limit 变量指定单个脚本程序可以使用的最大内存容量变量 memory_limit 的值应当适当大于 post_max_size 的值。
- max_execution_time：设置了在强制终止脚本前 PHP 等待脚本执行完毕的时间，此时间以秒计算。当脚本进入了一个无限循环状态时此变量非常有用。然而，当存在一个需要很长时间完成的合法活动时（如上传大尺寸文件）这项功能也会导致操作失败。在这样的情况下，必须考虑将此变量值增加，以避免 PHP 在脚本正在执行某些重要过程的时候将脚本关闭。

了解各项参数的含义后开始调节相关项目的参数，打开 php.ini，需要修改如下几处。

- file_uploads =on：是否允许通过 HTTP 上传文件的开关，默认为 ON，即开。
- upload_tmp_dir：文件上传至服务器上存储临时文件的地方，如果没指定就会用系统默认的临时文件。
- upload_max_filesize = 8m：允许上传文件大小的最大值，默认为 2M。
- post_max_size = 8m：指通过表单 POST 给 PHP 的所能接收的最大值，包括表单里的所有值，默认为 8M。

设置好上述 4 个参数后，可以上传尺寸小于 8M 的文件。如果需要上传尺寸更大的文件，则需要进一步配置以下参数。

- max_execution_time = 600：每个 PHP 页面运行的最大时间值（秒），默认为 30 秒。
- max_input_time = 600：每个 PHP 页面接收数据所需的最大时间，默认为 60 秒。
- memory_limit = 8m：每个 PHP 页面所占用的最大内存值，默认为 8MB。

对于 Linux 主机，还需要调整 php.conf 文件中的 LimitRequestBody 524288 将 524 288（512×1024）改大，如改为 5M（5×1024×1024），这样上传就不会失败。

7.3　数据库设计

用户上传的文件资料（文件可以是一个软件、一个文档资料、图片），当这些用户自主上传的文件增多后，涉及文件归档和整理的问题。合理归档后的文件资料，会方便更多的访问者使用，同时也便于管理者管理。

上传与下载模块数据库设计并不复杂，核心只有一个“上传文件资料表”，该表被用来记录上传文件的详细信息，同时可以提供查询。该数据库设计属于标准的业务表结构设计，读者在理解含义的基础上可以应用于很多业务模型。下面来看具体的设计步骤。

（1）建立 download 数据库，SQL 语句如下：

```
CREATE DATABASE 'download' ;
```

（2）在 download 数据库中建立一个命名为 f_detail 的“上传文件资料表”，结构如表 7.1 所示。创建表及相关字段的 SQL 语句如下：

```
CREATE TABLE 'f detail' (
'id' INT( 11 ) NOT NULL AUTO_INCREMENT PRIMARY KEY ,
'filename' VARCHAR( 100 ) NOT NULL ,
'des' VARCHAR( 64 ) NOT NULL ,
'fsize' VARCHAR( 10 ) NOT NULL ,
'ftype' VARCHAR( 100 ) NOT NULL ,
'utime' DATETIME NOT NULL
) ENGINE = MYISAM ;
```

表 7.1　上传文件资料表

字　段	数据类型	长　度	NULL	默认值	字段说明
id	int	11	否		自增主键
filename	varchar	100	否	无	文件名称
des	varchar	64	否	无	文件描述
fsize	varchar	10	否	无	文件尺寸
ftype	varchar	100	否	无	文件类型
utime	datetime		否		上传时间

注意：随着上传文档的增多，可以继续添加分类 id 等字段、授权方式、下载次数、运行环境等字段。这样就可以将整理后的文件发布出来，做成一个下载频道提供用户下载了。

7.4　文件上传程序

PHP 实现文件上传的原理已经在上面做了详细介绍，本节将对文件上传程序做必要的扩展，实现完整的文件上传程序。其中，upload.html 文件是客户端程序，用来存储上传表单的 JavaScript 代码；upload.php 文件是服务端的文件上传程序，用来处理客户端提交过来

的上传文件。

7.4.1　上传文件的客户端类型检测

文件上传程序的作用是对客户端提交的文件类型进行检测和过滤，这样可以有效地保护系统安全，类型检测代码如下：

```
header("content-Type: text/html; charset=gb2312");
$uptypes=array('image/jpg',            //上传文件类型列表
 'image/jpeg',
 'image/png',
 'image/pjpeg',
 'image/gif',
 'image/bmp',
 'application/x-shockwave-flash',
 'image/x-png',
 'application/msword',
 'audio/x-ms-wma',
 'audio/mp3',
 'application/vnd.rn-realmedia',
 'application/x-zip-compressed',
 'application/octet-stream');
//文件参数条件
$max_file_size=20000000;               //上传文件大小限制，单位 BYTE
$path_parts=pathinfo($_SERVER['PHP_SELF']); //取得当前路径
$destination_folder ="up/";            //上传文件路径
$imgpreview=1;                         //是否生成预览图(1 为生成，其他为不生成)
$imgpreviewsize=1/2;                   //缩略图比例
```

【代码解读】

上面这段代码定义了允许上传的文件类型，也定义了其他辅助变量，如文件尺寸显示变量、系统路径变量、缩略图预览等变量。

7.4.2　上传文件的服务器端处理

上传客户端文件（upload.html）中定义了 getFileSize(filename)函数，它的作用是对上传的文件尺寸在客户端进行验证。由于客户端程序的各种限制，不能对上传文件做更多地判断和检测，这部分工作只能交给服务器端的 PHP 脚本文件来完成。

现在打开 upload.php 文件，完成类型判断、异常处理、信息提示、缩略图（仅限图片文件）等功能，核心代码如下：

```
/*
*功能：上传文件检测程序
*/
$file = &$HTTP_POST_FILES ['userfile'];
if ($max_file_size < $file ["size"]){          //检查文件大小
    echo "<font color='red'>文件太大! </font>";
    exit ();
}
if (! in_array ( $file ["type"], $uptypes ))   //检查文件类型
{
```

```
    echo "<font color='red'>不能上传此类型文件！</font>";
    exit ();
}
//检查上传目录是否存在，如果不存在则创建
if (! file_exists ( $destination_folder )) {
    mkdir ( $destination_folder );
}
$filename = $file ["tmp_name"];
$image_size = getimagesize ( $filename );
$pinfo = pathinfo ( $file ["name"] );
$ftype = $pinfo [extension];
$destination = $destination_folder . time () . "." . $ftype;
$fname = time () . "." . $ftype;
if (file_exists ( $destination ) && $overwrite != true) {
    echo "<font color='red'>同名文件已经存在了！</a>";
    exit ();
}
//移动文件异常处理
if (! move_uploaded_file ( $filename, $destination )) {
    echo "<font color='red'>移动文件出错！</a>";
    exit ();
} else {
    //添加写入数据库的部分
    //创建数据库连接
    $con = mysql_connect ( 'localhost', 'root', '198251' ) or die ( 'Could
    not connect: ' . mysql_error () );
    //echo 'Connected successfully';
    $db = mysql_select_db ( 'download', $con );
    if (! $db) {
        die ( "Can\'t use download : " . mysql_error () );
    } else {
        //将用户信息插入数据库的 user 表
        $sql = "INSERT INTO 'download'.'f_detail' ('id' ,'filename',
        'des' ,'fsize' ,'ftype' ,'utime' )VALUES (NULL ,'" . $fname . "' ,
        '', '" . $file ["size"] . "', '" . $file ["type"] . "',NOW());";
        echo $sql;
        $result = mysql_query ( $sql );
        if (! $result) {
            mysql_free_result ( $result ); //释放结果集
            mysql_close ( $db );           //关闭连接
            echo '数据记录插入失败!';
            exit ();
        }
    }
}
$pinfo = pathinfo ( $destination );
$fname = $pinfo [basename];
echo "<table><tr><td>上传文件地址：http://" . $_SERVER ['SERVER_NAME'] .
$path_parts ["dirname"] . "/" . $destination_folder . $fname . "</td>
</tr></table>";
echo " 宽度:" . $image_size [0];
echo " 长度:" . $image_size [1];
if ($watermark == 1) {
    $iinfo = getimagesize ( $destination, $iinfo );
    $nimage = imagecreatetruecolor ( $image_size [0], $image_size [1] );
    $white = imagecolorallocate ( $nimage, 255, 255, 255 );
    $black = imagecolorallocate ( $nimage, 0, 0, 0 );
    $red = imagecolorallocate ( $nimage, 255, 0, 0 );
    imagefill ( $nimage, 0, 0, $white );
```

```
    switch ($iinfo [2]) {
        case 1 :
            $simage = imagecreatefromgif ( $destination );
            break;
        case 2 :
            $simage = imagecreatefromjpeg ( $destination );
            break;
        case 3 :
            $simage = imagecreatefrompng ( $destination );
            break;
        case 6 :
            $simage = imagecreatefromwbmp ( $destination );
            break;
        default :
            die ( "<font color='red'>不能上传此类型文件! </a>" );
            exit ();
    }
    imagecopy ( $nimage, $simage, 0, 0, 0, 0, $image size [0], $image size
    [1] );
    imagefilledrectangle ( $nimage, 1, $image size [1] - 15, 80, $image size
    [1], $white );
    switch ($iinfo [2]) {
        case 1 :
            imagejpeg ( $nimage, $destination );
            break;
        case 2 :
            imagejpeg ( $nimage, $destination );
            break;
        case 3 :
            imagepng ( $nimage, $destination );
            break;
        case 6 :
            imagewbmp ( $nimage, $destination );
            break;
    }
    //覆盖原上传文件
    imagedestroy ( $nimage );
    imagedestroy ( $simage );
}
if ($imgpreview == 1) {
    echo "<br>图片预览:<br>";
    echo "<a href=\"" . $destination . "\" target=' blank'><img src=\"" .
    $destination . "\" width=" . ($image_size [0] * $imgpreviewsize) . "
    height=" . ($image_size [1] * $imgpreviewsize);
    echo " alt=\"图片预览:\r 文件名:" . $fname . "\r 上传时间:" . date ( 'm/d/Y
    h:i' ) . "\" border='0'></a>";
}
?>
```

修改后的 upload.php 程序可以完成将文件上传到指定的目录中的功能。当然，现在只是将文件上传到服务器的制定目录中去了，这样的文件凌乱且没有意义。

7.4.3　将上传文件信息归档到数据库

上一节已经在数据库中创建了 f_detail 表，用来将上传文件信息做归档和记录。下面写一段 SQL 语句，作用是当文件上传成功时向数据库中写入文件的信息，SQL 语句如下：

```
INSERT INTO 'download'.'f detail' ('id' ,'filename' ,'des' ,'fsize' ,
'ftype' ,'utime' )
VALUES (NULL , $filename, $des, $fsize, $ftype,$utime);
```

下面打开 upload.php 文件，将信息写入部分代码添加到程序中，需要添加的代码如下：

```
<?php
if (! move uploaded file ( $filename, $destination )) {
    echo "<font color='red'>移动文件出错！</a>";
    exit ();
} else {
    //添加写入数据库的部分
    //创建数据库连接
    $con = mysql connect ( 'localhost', 'root', '198251' ) or die ( 'Could
    not connect: ' . mysql error () );
    //echo 'Connected successfully';
    $db = mysql select db ( 'download', $con );
    if (! $db) {
        die ( "Can\'t use download : " . mysql error () );
    } else {
        //将用户信息插入数据库的 f_detail 表
$sql = "INSERT INTO 'download'.'f detail' ('id' ,'fname' ,'des' ,'fsize',
'ftype' ,'utime' )VALUES (NULL ,'" . $destination . "' , '', '" . $file
["size"] . "', '" . $file ["type"] . "',NOW());";
        $result = mysql_query ( $sql );
        if (! $result) {
            mysql_free_result ( $result ); //释放结果集
            mysql close ( $db );           //关闭连接
            echo '数据记录插入失败!';
            exit ();
        }
    }
}
?>
```

注意：SQL 语言中有很多内部函数，例如上面的代码中出现的 now()函数，其含义是在获得当前系统时间对应数据库字段的 datetime()类型，这样就减少了用 PHP 来处理时间的步骤，提高了效率。

通过以上 3 个步骤，完成了一个完整的文件上传程序。读者可以根据自己的需要对这个程序进行改造，添加更多的功能。这个核心程序所展示的内部流程是任何与上传相关的程序共同的设计思路。

7.5　文件下载程序

经过前几节的学习，读者应该对文件上传下载的逻辑过程和技术实现有了基本的认识。本节将完成已上传文件的列表展示功能和对应文件的下载。文件下载部分程序由两部分构成，即文件列表页面（list.php）和文件下载程序（download.php）组成。

7.5.1　文件下载列表

文件下载页面用来将已经上传的文件以列表的方式直观地展示出来，方便下载者了解所下载文件的基本信息。下面设计一个基本的下载页面，核心框架部分代码如下：

```
<HEAD>
<TITLE>文件下载列表</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="base.css" type=text/css rel=stylesheet>
<BODY>
<DIV class=bodytitle>
<DIV class=bodytitleleft></DIV>
<DIV class=bodytitletxt>文件下载列表</DIV>
</DIV>
<TABLE class=tbtitle style="MARGIN-TOP: 6px; BACKGROUND: #e2f5bc"
    height=31 cellSpacing=1 cellPadding=1 width="96%" align=center
    border=0>
    <TBODY>
        <TR>
            <TD class=tbtitletxt><STRONG> 请选需要下载的文件：</STRONG>
            </TD>
        </TR>
    </TBODY>
</TABLE>
<TABLE class=tblist
    style="BORDER-RIGHT: #e2f5bc 1px solid; BORDER-TOP: #e2f5bc 1px solid;
    MARGIN: 0px 0px 6px; BORDER-LEFT: #e2f5bc 1px solid; BORDER-BOTTOM:
    #e2f5bc 1px solid"
    cellSpacing=1 cellPadding=1 width="96%" align=center border=0>
    <TBODY>
        <TR align=middle>
            <TD class=tbsname
                style="PADDING-RIGHT: 6px; PADDING-LEFT: 6px; PADDING-
                BOTTOM: 20px; PADDING-TOP: 6px"
                vAlign=top align=left height=120>
            <TABLE cellSpacing=0 cellPadding=6 width="90%" border=0>
                <TBODY>
                    <TR>
                        <TD><STRONG>下载说明：</STRONG><BR>
                        1、选择需要下载的文件后在'操作选项'中选择'文件下载'选项；
                        </TD>
                    </TR>
                </TBODY>
            </TABLE>
            <TABLE cellSpacing=1 cellPadding=3 width="98%" bgColor=#cae886
                border=0>
                <TBODY>
                    <TR align=middle height=20>
                        <TD width="8%" bgColor=#dcf4bd><STRONG>ID
                        </STRONG></TD>
```

```
                        <TD width="17%" bgColor=#dcf4bd><STRONG>文件名
                        </STRONG></TD>
                        <TD width="11%" bgColor=#dcf4bd><strong>文件尺寸
                        </strong></TD>
                        <TD width="18%" bgColor=#dcf4bd><strong>文件类型
                        </strong></TD>
                        <TD width="18%" bgColor=#dcf4bd><strong>上传时间
                        </strong></TD>
                        <TD width="28%" bgColor=#dcf4bd><STRONG>操作选项
                        </STRONG></TD>
                    </TR>
                    <TR onMouseMove="JavaScript:this.bgColor='#FCFDEE';"
                        onmouseout="JavaScript:this.bgColor='#FFFFFF';"
                        align=middle
                        bgColor=#ffffff height=20>
                        <TD>1</TD>
                        <TD>图书目录.doc</TD>
                        <TD>10kb</TD>
                        <TD>.doc</TD>
                        <TD>0</TD>
                        <TD><A href="http://localhost/dedecms/dede/
                        catalog main.php">文件下载</A>
                        | <FONT color=#cccccc>扩展功能 1</FONT> | <FONT
                        color=#cccccc>扩展功能 2</FONT></TD>
                    </TR>
                </TBODY>
            </TABLE>
            <TABLE cellSpacing=0 cellPadding=0 width="90%" border=0>
            </TABLE>
            </TD>
        </TR>
    </TBODY>
</TABLE>
</BODY>
```

接下来在数据库中将已经下载的文件提取出来，再嵌套到 HTML 页面中。经过前面的学习，这段代码读者应该很容易写出来，展示部分的核心代码（页面模板部分代码请读者参考源程序）如下：

```
<TBODY>
    <TR align=middle height=20>
        <TD width="8%" bgColor=#dcf4bd><STRONG>ID</STRONG></TD>
        <TD width="17%" bgColor=#dcf4bd><STRONG>文件名</STRONG></TD>
        <TD width="11%" bgColor=#dcf4bd><strong>文件尺寸</strong></TD>
        <TD width="18%" bgColor=#dcf4bd><strong>文件类型</strong></TD>
        <TD width="18%" bgColor=#dcf4bd><strong>上传时间</strong></TD>
        <TD width="28%" bgColor=#dcf4bd><STRONG>操作选项</STRONG></TD>
    </TR>
        <?php
        $con = mysql connect ( 'localhost', 'root', '198251' ) or die ( 'Could
        not connect: ' . mysql_error () );
        //echo 'Connected successfully';
        $db = mysql select db ( 'download', $con );
        if (! $db) {
            die ( "Can\'t use download : " . mysql error () );
        } else {
            $sql = "SELECT * FROM 'f detail'";
            $result = mysql_query ( $sql, $con );
```

```
            $err = mysql error ();
            if ($err) {
                echo "发生错误，请联系管理员";
            }
            while ( $row = mysql fetch array ( $result, MYSQL NUM ) ) {
            ?>
        <!--循环 start-->
    <TR align=middle bgColor=#ffffff height=20>
        <TD><?php  echo $row [0]?></TD>
        <TD><?php  echo $row [1]?></TD>
        <TD><?php  echo $row [3]?></TD>
        <TD><?php  echo $row [4]?></TD>
        <TD><?php  cho $row [5]?></TD>
        <TD><A href="download.php?id=<?php
                echo $row [0]?>">文件下载</A> | <FONT
            color=#cccccc>扩展功能 1</FONT> | <FONT color=#cccccc>扩展功能
            2</FONT></TD>
    </TR>
    <!--循环 end-->
        <?php
            }
            mysql_free_result ( $result );
        }
        ?>
</TBODY>
```

【代码解读】

在上述代码中，将数据中的文件上传信息读取出来放入数组中，然后在通过 while()函数循环遍历出来，再将输出的内容嵌入页面代码模板中。

运行 list.php 文件，执行结果如图 7.3 所示。从图中可以看到，程序已经正确地显示了文件列表和相应的下载链接。



图 7.3　文件下载列表页面

7.5.2　文件下载程序实现

打开 download.php 文件，该文件是负责处理文件下载逻辑的。但是它不完善，现在要根据“文件下载列表页面”中“文件下载”选项提供的链接形式接收参数。

注意：在使用 header()函数发送原始 HTTP 标头时，该文件不能有任何字符的输出，否则将影响浏览器识别导致下载失败。

该文件的具体的代码如下：

```
<?
//创建数据库连接
$con = mysql connect ( 'localhost', 'root', '198251' ) or die ( 'Could not
connect: ' . mysql_error () );
//echo 'Connected successfully';
$db = mysql select db ( 'download', $con );
if (! $db) {
    die ( "Can\'t use download : " . mysql error () );
} else {
    //将用户信息插入数据库的 user 表
    $sql = "SELECT * FROM 'f_detail' WHERE 'id' ='" . $_GET ['id'] . "' LIMIT
    0 , 30";
    $result = mysql query ( $sql, $con );
    $row = mysql_fetch_row ( $result );
    if (! $result) {                                    //释放结果集
        mysql_free_result ( $result );                  //关闭连接
        mysql close ( $db );
        echo '数据记录插入失败!';
        exit ();
    }
}
$file_name = $row [1];          //下载文件名形式如：$file_name = "xxx.rar"
$file dir = "up/";              //下载文件存放目录
if (! file_exists ( $file_dir . $file_name )) {//检查文件是否存在
    echo "文件找不到";
    exit ();
} else {
    $file = fopen ( $file_dir . $file_name, "r" ); //打开目标文件
    //输入文件标签
    Header ( "Content-type: application/octet-stream" );
    Header ( "Accept-Ranges: bytes" );
    Header ( "Accept-Length: " . filesize ( $file dir . $file name ) );
    Header ( "Content-Disposition: attachment; filename=" . $file name );
    //输出文件内容
    //读取文件内容并直接输出到浏览器
    echo fread ( $file, filesize ( $file dir . $file name ) );
    fclose ( $file );
    exit ();
}
?>
```

执行 download.php 这个文件，给定 id 参数为 3，根据数据库中对应的数据值可知道，该程序将提示下载名称为 1264272773.doc 的文件。执行文件效果如图 7.4 所示。

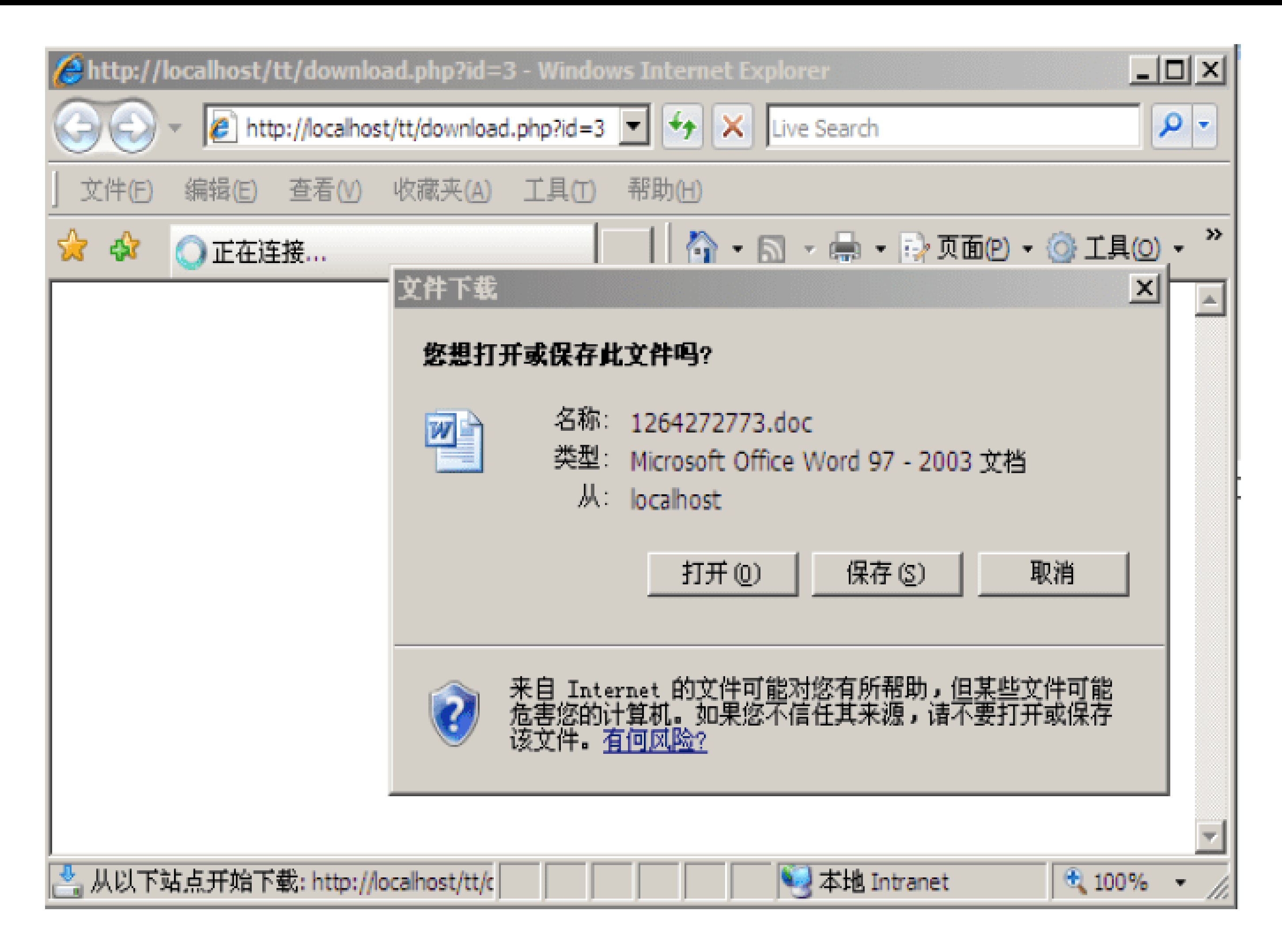


图 7.4　下载提示页面

至此，文件上传和下载的程序实现就全部完成了。通过前面的验证可以知道，程序达到了预先期望的文件归档和安全下载的目的。

7.6　小　　结

本章的重点是介绍 PHP 实现文件上传和下载的基本原理。文件上传与下载是网站的常用功能，它对数据的可靠性和安全性都有一定的要求。第 7.1 节讲解了文件上传与下载的原理。第 7.2 节介绍了文件尺寸限制及异常处理等，具体内容是客户端的验证解决方案（JavaScript 文件尺寸验证）和服务端的验证解决方案（PHP 上传文件涉及的参数）。在实现了文件的上传和下载之后，通过数据库对上传的文件进行归档和管理。第 7.5 节对整个上传和下载做了流程上的梳理，最终构成了完整的文件上传和下载的模块应用，并在这个过程中将遇到的技术难点做了详细的讲解。

第 8 章　内容自动采集器模块（爬虫+FCKeditor 编辑器+任务接力）

网站内容的组成除了编辑录入原创内容外，还可以通过后台的内容采集功能获取指定内容源信息。使用内容自动采集器可以使得网站的内容多样化，同时减少网站编辑人员的工作量。因此，内容自动采集器功能是网站后台的必要功能之一。读者通过本章的学习，可以了解内容采集的基本原理和实现方法，直观地了解正则表达式在内容采集过程中的核心作用，其中涉及的技术细节和知识点也将在讲解例子的时候逐一阐述。

本章主要涉及的知识点如下。

- file_get_contents()函数：函数把整个文件读入一个字符串中。
- preg_match_all()函数：进行全局正则表达式匹配。
- FCKeditor 编辑器：被广泛使用的、开放源代码的“所见即所得”文字编辑器。
- 任务接力模式：任务接力模式的本质是对一个任务做拆分，将一个任务拆分成多个子任务来实现。

8.1　内容自动采集器功能及采集原理

内容采集，顾名思义就是对互联网的公共信息资源按照一定的要求进行自动的采集、过滤、整理，再按照一定的规则存入数据库中。根据这个目标可以看到，内容自动采集器的功能由 3 部分组成，即数据规则模型管理、采集节点管理、已下载内容管理。

在实际的应用中，根据不同的业务应用领域会对应强化某部分的功能。如果要采集的目标网站内容格式十分复杂，就要强化“数据规则模型管理”定制适用于不同类型站点的采集规则；如果需要经常更换采集的信息源，则需要强化“采集节点管理”和“重复内容过滤”功能。常见的典型应用则是将上述的功能整合到一起，在一个界面上操作会更高效、快捷。总之，需要根据实际业务将上述的功能结合使用。

内容自动采集器采集数据的一般流程如下：

（1）根据预先定义的抓取规则，获取一个栏目的网页中的所有内容，需要先将这个网页的网址列表记录下来制作成采集列表。

（2）程序根据定义的规则抓取列表页面，从中分析整理出网址列表，然后再去抓取获得网址的网页中的内容。

（3）程序根据具体页面的采集规则，对下载到的网页分析，将标题内容等信息分离开，核对无误后存入数据库。

本节具体介绍内容采集技术实现的原理和流程。

8.1.1　URL 地址列表分析

待采集列表页面形式可以分为单页网址、多页网站、复合页网址 3 种。根据页面的不同类型采用不同的流程控制，具体的流程控制代码如下：

```
<?php
switch (URL) {
    case 1 :
        //单页的设置
        $url [0] = URL;
        break;
    case 2 :
        //多页的设置
        $list_content = file_get_contents ( $row->url );
        //观察链接特征：<li> <a href="/news/2010-05-21/1705605024 4.shtml"
        title="自主的胜利？" target="_blank" class="fl">自主的胜利？</a><i>
        //获取列表正则规则
        $list match = $row [2];
        preg match all ( $list match, $list content, $list data );
        break;
        case3
        //其他类型的网址观察具体的链接特征应用不同的正则规则
        break;
}
?>
```

【代码解读】

上述代码通过 switch()函数判断链接的类型，然后根据链接“特征”匹配正则表达式的规则获取指定的内容，这里可以将 switch()函数看作程序的流程控制器。

8.1.2　获得内容页面信息

当获得待下载的列表页面后，就可以提取需要的信息了。该程序功能最重要的就是提取正则表达式的部分，因为只要获得部分采集者认为有意义的信息即可。内容获取部分的流程代码如下：

```
<?php
//功能：获得内容页面信息
foreach ( $list data [1] as $detail url ) {
    $handles = @file_get_contents ( $detail_url ); //载入 URL 地址页
    preg match all ("/".链接规则."/is",$handles,$content match);
                                                    //正则表达式匹配链接规则
    INSERT INTO 'links' ( 'title' , 'url' , 'rules' , 'date' )
                                         //所得内容页面链接地址写入数据库 Link 表
}
?>
```

【代码解读】

其中，变量$list_data[1]代表已获得的列表数组，$content_match 代表正则表达式匹配链接内容的规则。其中核心的部分是两个函数，即 file_get_contents()和 preg_match_all()。preg_match_all()是正则函数，用来安装正则表达式定制的规则并提取相应的内容。

🔔注意：file_get_contents()函数是远程读取网页内容的，需要在 PHP5 以上的版本才能使用。

8.1.3　延时函数

由于 PHP 脚本有 30 秒的执行限制，当执行时间大于 30 秒时，系统会报一个超时的“致命级”错误终止程序运行，所以在抓取大量内容时需要应用下面的延时函数。

```
<?php
//功能：延时函数
$ch = curl_init ();
//设置超时时间
$timeout = 100;
curl_setopt ( $ch, CURLOPT_URL, $url );
curl_setopt ( $ch, CURLOPT_RETURNTRANSFER, 1 );
curl_setopt ( $ch, CURLOPT_CONNECTTIMEOUT, $timeout );
$handles = curl_exec ( $ch );  //执行句柄
curl_close ( $ch );            //关闭连接
?>
```

【代码解读】

上述函数的核心是 curl_setopt()函数，它将为一个 CURL 会话设置选项。该函数的描述如下：

```
bool curl_setopt (int ch, string option, mixed value)
```

其中，option 参数是具体的操作设置，value 是这个选项给定的数值。

8.1.4　获取全部列出形式

在执行以上的步骤后，程序已经获得了所需的内容信息。在对页面进行分页分析整理后全部列出形式，核心流程代码如下：

```
<?php
//正则匹配分页区域
preg_match ("/".分页规则."/is",$buffer, $regs2);
//查找每个分割里面的网址 for 保证网址不重复
preg_match_all
("/".'<[^<>]*(href|value)=(\"|\')?([^\'\"<>]*)(\"|\')?[^<>]*>'."/is",$re
gs2[1],$regs3);
for($i = 0; $i <= count ( $regs3 [3] ); $i ++) {
    $gethttp = $string->gethttp ( $regs3 [3] [$i] );
    $buffer2 = @file_get_contents ( $gethttp );
    //延时
    if (empty ( $buffer2 )) {
        if ($phpcurl_init == yes) {
            $ch = curl_init ();
            $timeout = 10;                  //设置延时时间
            curl_setopt ( $ch, CURLOPT_URL, $regsar [$i] );
            curl_setopt ( $ch, CURLOPT_RETURNTRANSFER, 1 );
            curl_setopt ( $ch, CURLOPT_CONNECTTIMEOUT, $timeout );
            $buffer2 = curl_exec ( $ch );  //执行句柄
            curl_close ( $ch );            //关闭连接
```

```
        }
    preg_match (”/”.内容规则.”/is”,$buffer2, $regss);
        $cont .= $regss [1];
    }
}
?>
```

【代码解读】

上述代码通过分页的规则匹配出分页列表，然后通过查找每个分割单元里的网址（for 函数排除重复网址），从而获得全部列出形式的列表数组。

8.1.5　分页分析

当文章内容过多时通常都会采用分页的形式，这样保证了页面的整体美观。但是对于采集程序而言就需要多做一步分页的分析工作，程序根据不同的分页样式分析出上下页形式。分页分析的流程代码如下：

```
<?php
if(preg match ("/".分页规则."/is",$buffer, $ljregs))
{
    while ( $ljregs [1] != "" ) {
        $ljregs [1] = $string->gethttp ( $ljregs [1] );
        $buffer = @file get contents ( $ljregs [1] );
        //延时
        if (empty ( $buffer )) {
            if ($phpcurl init == "yes") {
                $ch = curl init ();
                $timeout = 10;                  //设置超时时间
                curl_setopt ( $ch, CURLOPT_URL, $ljregs [1] );
                curl setopt ( $ch, CURLOPT RETURNTRANSFER, 1 );
                curl_setopt ( $ch, CURLOPT_CONNECTTIMEOUT, $timeout );
                $buffer = curl exec ( $ch );    //执行句柄
                curl_close ( $ch );             //关闭连接
            }
        }
        preg match ( "/" . $body rule . "/is", $buffer, $regs );//匹配规则
        $cont .= $regs [1];
    }
}
?>
```

【代码解读】

上述代码根据获得分页规则的不同，匹配不同的上下页形式，通过 while()函数嵌套 if()函数，对获得内容过程中的异常状态进行排除处理。

注意： 在 file_get_contents ()函数前使用@符号，可以抑制程序报错时的输出。

8.2　数据库设计

内容自动采集器模块的数据库设计目的是满足两个需求：一个是规则存储表（spiders）存储采集规则、采集源 URL 地址、栏目分类；另一个是采集内容表（articles）用来存储采

集到的数据。

8.2.1　规则存储表设计

规则存储表（spiders）是内容采集模块的核心，它由两个功能部分组成，一部分是存储采集规则（页面网址提取规则、页面内容提取规则）；另一部分是辅助功能（栏目名称、已采集数、栏目分类标识）。了解清楚功能划分之后，下面来看具体的设计步骤。

（1）建立 get_content 数据库，SQL 语句如下：

```
CREATE DATABASE 'get_content ;
```

（2）在 get_content 数据库中建立一个命名为 spiders 的“规则存储表”，结构如表 8.1 所示。创建表及相关字段的 SQL 语句如下：

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
--
-- 数据库: 'get_content'
--
-- --------------------------------------------------------
--
-- 表的结构 'spiders'
--
CREATE TABLE 'spiders' (
  'ID' int(10) unsigned NOT NULL auto_increment,
  'Title' varchar(200) NOT NULL,
  'ListPreg' mediumtext NOT NULL,
  'ContentPreg' mediumtext NOT NULL,
  'Count' int(10) unsigned NOT NULL default '0',
  'EnterUrl' varchar(200) NOT NULL,
  'Category' smallint(5) unsigned NOT NULL,
  PRIMARY KEY  ('ID')
) ENGINE=MyISAM ;
```

表 8.1　规则存储表

字　段	数据类型	长　度	NULL	默认值	字段说明
ID	int	10	否	auto_increment	自增主键
Title	varchar	200			栏目名称
ListPreg	mediumtext				页面网址提取规则
ContentPreg	mediumtext				页面内容提取规则
Count	int	10	否		已采集数
EnterUrl	varchar	200			采集源 URL 地址
Category	smallint	5			分类标识

注意： 以上字段是一个采集规则表的基础，根据业务的扩展可以添加用于标识 VIP 会员的 vip 字段等。

8.2.2　采集内容表设计

采集内容表（articles）的作用是存储采集到的数据，从命名就可以看出它的设计和一般文章内容表有很多相似的地方。这部分的数据库设计相对更简单些，只有一个“采集内

容表”，它被用来记录采集到的文件的详细信息，同时可以提供查询。但是这类表的设计在其他项目中也很常见，适用于多种业务模型。读者在理解含义的基础上可以做适当的扩展，使之适用于更多的业务模型。

在 get_content()数据库中建立一个命名为 articles 的“采集内容表”，如表 8.2 所示。创建表及相关字段 SQL 语句如下：

```
SET SQL MODE="NO AUTO VALUE ON ZERO";
--
-- 数据库: 'get_content'
--
-- --------------------------------------------------------
--
-- 表的结构 'articles'
--
CREATE TABLE 'articles' (
  'ID' int(10) unsigned NOT NULL auto increment,
  'Title' varchar(200) NOT NULL,
  'Click' int(11) NOT NULL,
  'Content' longtext,
  'Date' date NOT NULL,
  'Category' smallint(6) NOT NULL,
  PRIMARY KEY  ('ID'),
  KEY 'Category' ('Category')
) ENGINE=MyISAM default charset=utf8;
```

表 8.2　采集内容表

字　　段	数 据 类 型	长　　度	NULL	默　认　值	字 段 说 明
ID	int	11	否		自增主键
Title	varchar	200	否	无	文章标题
Click	int	11	否	无	文章内容中包含的链接
Content	longtext		否	无	记录页面内容
Date	date		否	无	采集日期
Category	smallint	6	否		栏目分类标识

8.3　采集器模块界面设计

采集器模块设计到界面设计共有两个部分，按功能分为内容自动采集器管理界面和采集内容列表页面。

8.3.1　采集器管理界面

采集器管理管理界面通常由采集器描述、所属栏目、列表页面地址、列表页匹配描述、内容页匹配描述组成。采集器管理管理界面的 HTML 核心功能设计代码如下：

```
<table align=center cellspacing=0 cellpadding=0>
    <form method="post">
    <tr class=head>
        <td colspan="2">采集器管理</td>
    </tr>
```

```
    <tr class=line>
        <td width="11%">采集器描述</td>
        <td width="89%">
        <input name="Title" type="text" class="input"
            id="Title" size="40" value="" /></td>
    </tr>
    <tr class=line>
        <td width="11%">所属栏目</td>
        <td width="89%"><select name="Category">
            <option value="1">汽车频道</option>
        </select></td>
    </tr>
    <tr class=line>
        <td valign="top">列表页面地址</td>
        <td><input name="EnterUrl" type="text" class="input"
            value="列表页面地址" size="60" /></td>
    </tr>
    <tr class=line>
        <td valign="top">列表页匹配描述</td>
        <td><textarea name="ListPreg" cols="80" rows="4" class="input">列
        表页匹配描述</textarea></td>
    </tr>
    <tr class=line>
        <td valign="top">内容页匹配描述</td>
        <td><textarea name="ContentPreg" cols="80" rows="4" class="input">
        内容页匹配描述</textarea></td>
    </tr>
    <tr class=line>
        <td colspan="2" align="center"><input name="Submit" type="submit"
            class="btn" value="提交" /> <input name="Submit" type="button"
            class="btn" value="返回" onclick="history.go(-1);" /></td>
    </tr>
    </form>
</table>
```

将上面的代码保存为 manage.php 文件，在浏览器中加载此文件，预览效果如图 8.1 所示。

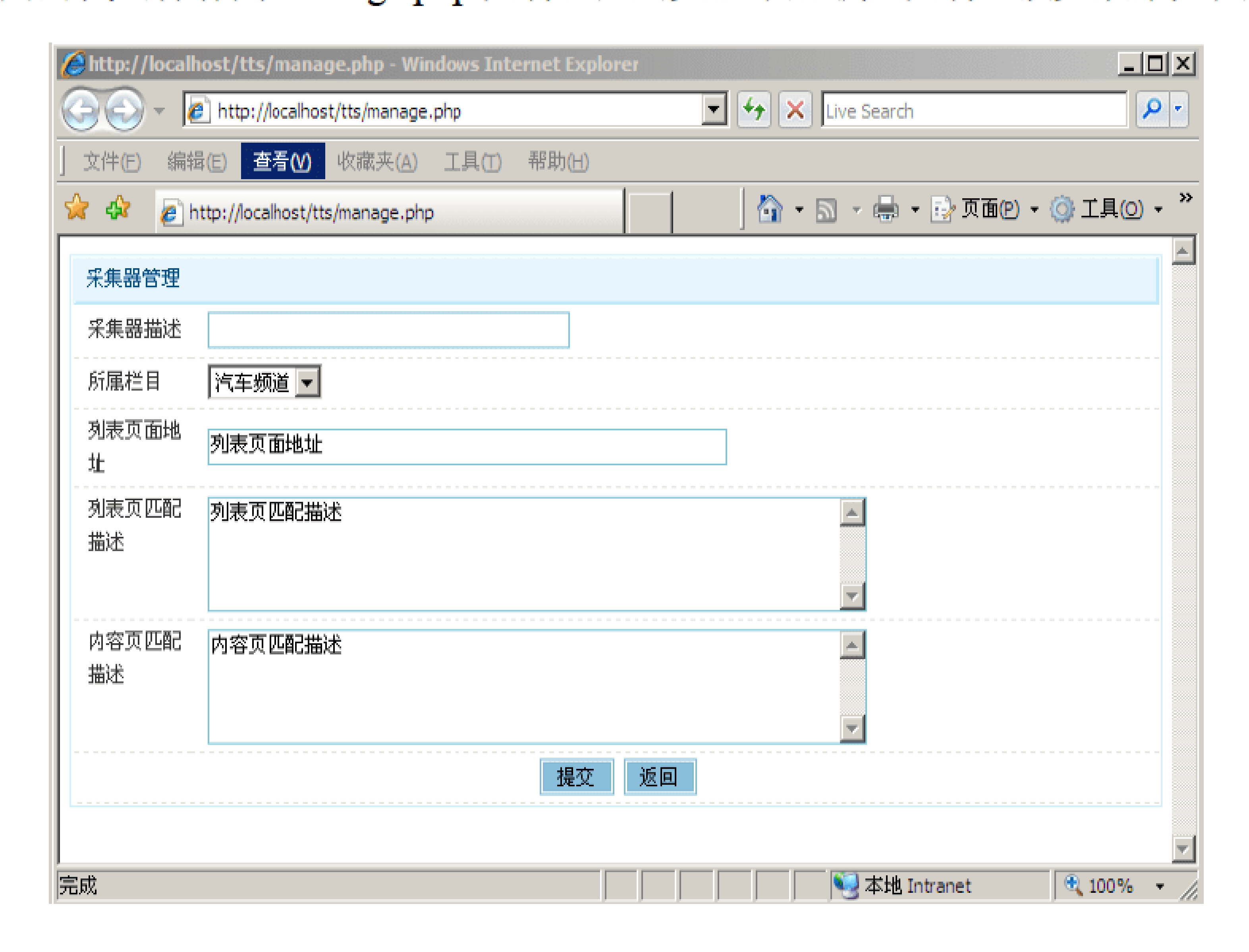


图 8.1　采集器管理界面

8.3.2　采集内容列表页面

采集内容列表页面的设计相对简单些，主要的功能是将采集到的信息，以列表的形式输出到浏览器中，该页面同时提供内容修改功能按钮，便于内容的维护。采集内容列表页面的 HTML 设计代码如下：

```
<table align=center cellspacing=0 cellpadding=0>
    <tr class=head>
        <td colspan="4">栏目名称</td>
    </tr>
    <tr class=line>
        <td><strong>日期</strong></td>
        <td width="913"><strong>标题</strong></td>
        <td><strong>修改</strong></td>
    </tr>
    <tr class=line>
        <td width="135">日期</td>
        <td>标题</td>
        <td width="83"><input name="Submit2" type="button" class="btn"
            value="修改" onclick="" /></td>
    </tr>
    <tr class=line>
        <td colspan="4" align="center"> </td>
    </tr>
</table>
```

将以上代码保存为 articles.php 文件，执行该文件，效果如图 8.2 所示。

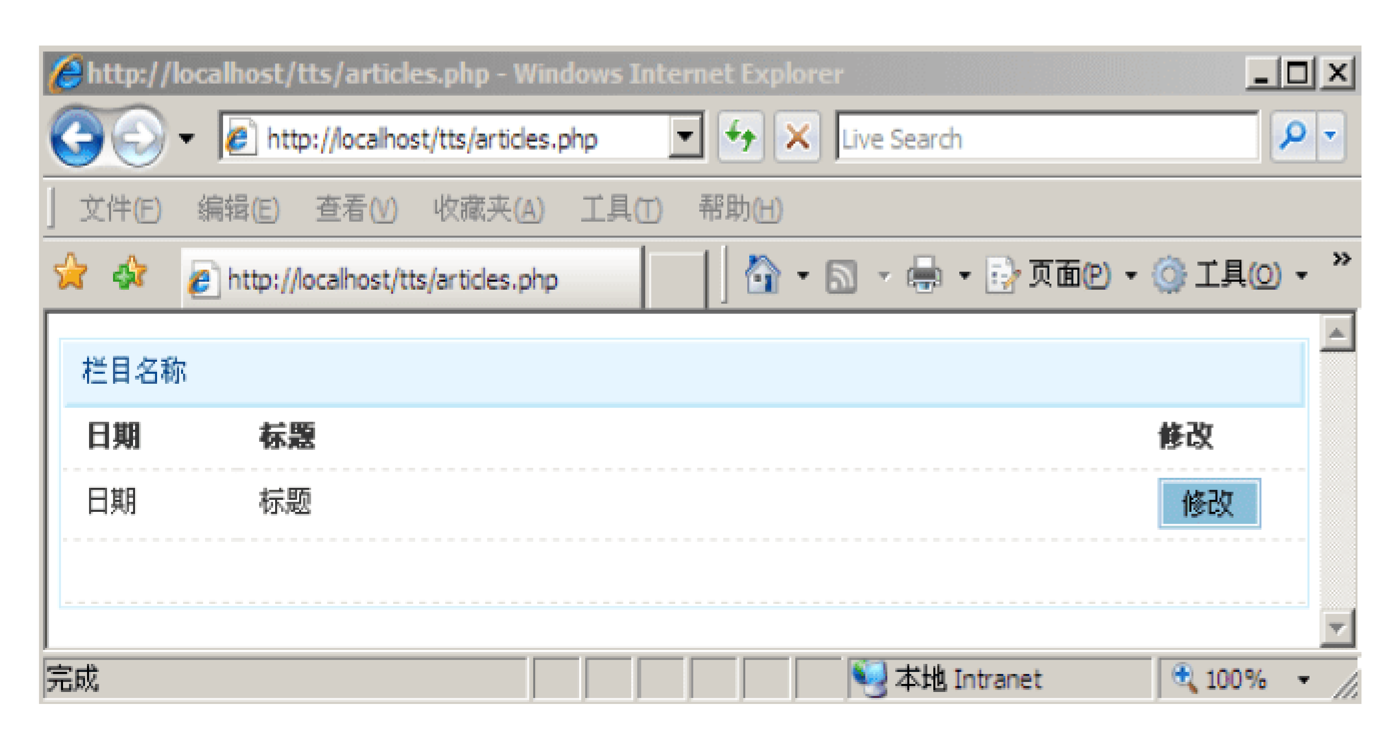


图 8.2　采集内容列表页面

8.4　内容自动采集器程序实现

采集器模块包含 3 个部分，即爬虫程序、采集规则管理程序、采集内容管理程序，这些功能模块共同完成采集内容的抓取功能。本节将会以前面介绍的技术原理为基础，详细地介绍这几个组成部分的逻辑管理，以及具体的编码过程，其中涉及的知识点也会逐一详细讲解。

8.4.1　采集规则管理程序

采集规则管理程序的核心是基于正则表达式的内容管理程序。了解如何使用正则表达式获得指定的内容信息，是采集规则管理程序的重点。

正则表达式的应用范围非常广泛，涉及的内容也十分庞杂，读者如果感兴趣可以自己对照手册学习，这里不作详细阐述。这里主要介绍正则表达式在内容采集方面的应用。采集器的主要工作是应用正则表达式对页面代码做匹配，进而获得相应的内容。这也是正则表达式在 Web 方面的主要应用。下面的例子是使用 PHP 正则表达式提取超链接及其标题，代码如下：

```
$str = '<a id="top8" href="http://list.mp3.baidu.com/song/A.htm?top8"
class="p14" target="_top">歌曲列表</a>
<br>
<a target="_blank" id="bp" href="http://list.mp3.baidu.com/list/bangping.
html" class="p14">中文金曲榜</a>
<br>
<td nowrap="nowrap">>#8226;>nbsp;<a id="top19" href="qingyinyue.html?
top19" class="p14" target="_top">轻音乐</a></td>';
$pat = '/<a (.*?)href="(.*?)"(.*?)>(.*?)<\/a>/i'; preg_match_all($pat,
$str,$m);
print_r($m);
```

执行上面一段代码，输出结果如下：

```
Array
(
    [0] => Array
        (
            [0] => <a id="top8" href="http://list.mp3.baidu.com/song/A.
            htm?top8" class="p14" target="_top">歌曲列表</a>
            [1] => <a target="_blank" id="bp" href="http://list.mp3.baidu.
            com/list/bangping.html" class="p14">中文金曲榜</a>
            [2] => <a id="top19" href="qingyinyue.html?top19" class="p14"
            target="_top">轻音乐</a>
        )
    [1] => Array
        (
            [0] =>  id="top8"
            [1] =>  target="_blank" id="bp"
            [2] =>  id="top19"
        )
    [2] => Array
        (
            [0] => http://list.mp3.baidu.com/song/A.htm?top8
            [1] => http://list.mp3.baidu.com/list/bangping.html
            [2] => qingyinyue.html?top19
        )
    [3] => Array
        (
            [0] =>  class="p14" target="_top"
            [1] =>  class="p14"
            [2] =>  class="p14" target="_top"
        )
    [4] => Array
```

```
        (
            [0] => 歌曲列表
            [1] => 中文金曲榜
            [2] => 轻音乐
        )
)
```

【代码解读】

通过上述代码可以了解到，通常的 PHP 正则表达式提取超链接获得内容的基本步骤。根据前面的需求分析可以知道，采集器模块涉及两个部分规则定制，即页面网址提取规则和页面内容提取规则。本章设定的抓取页面（http://auto.sina.com.cn/news/t/）是一个网站的汽车频道的列表页面。

URL 列表页面的链接样式如下：

```
<li> <a href="/news/2010-05-21/1705605024 4.shtml" title="自主的胜利？" target="_blank" class="fl">自主的胜利？</a><i>
```

通过观察链接的特征，书写如下正则表达式来获得匹配信息。

```
$list_match = '/<li> <a href="(\/news\/.+?\.shtml)" title=".+?" target="_blank" class="fl">/';
```

获得内容页面的正则表达式依据和上面的方法一样，通过观察内容页面模块的部分 HTML 代码，可以得到正则匹配规则如下：

```
/<!-- publish_helper(.+?) -->(.+?)<!-- publish_helper_end -->/is
```

现在打开 manage.php 文件添加保存规则部分的代码，下面的程序将上面的功能做了整合，这样就可以完成一个完整的功能，代码如下：

```
<?php
$con = mysql_connect ( 'localhost', 'root', '198251' ) or die ( 'Could not connect: ' . mysql_error () );
mysql_query ( "set names gb2312" ); //字符集设定
//echo 'Connected successfully';
$db = mysql_select_db ( 'get_content', $con );
if (! $db) {
    die ( "Can\'t use download : " . mysql_error () );
} else {
    $sql = "SELECT * FROM 'spiders' LIMIT 0 , 30  ;";
    $result = mysql_query ( $sql, $con );
    $row = mysql_fetch_array ( $result );
    if ($_POST ['flag'] == 1) {
        //更新定义规则
        $sql1 = "UPDATE 'get_content'.'spiders' SET 'ListPreg' = '" . $_POST
        ['ListPreg'] . "','ContentPreg' = '" . $_POST ['ContentPreg'] . "',
        'EnterUrl' = '" . $_POST ['EnterUrl'] . "' WHERE 'spiders'.'ID' =4
        LIMIT 1 ;";
        $result1 = mysql_query ( $sql1, $con );
        mysql_free_result($result1);
        mysql_close($db);//关闭连接
        header ( "location:manage.php" );
    }
}
?>
<table align=center cellspacing=0 cellpadding=0>
    <form method="post" action="manage.php"><input type="hidden"
```

```
        name="flag" value="1">

    <tr class=head>
        <td colspan="2">采集器管理</td>
    </tr>
    <tr class=line>
        <td width="11%">采集器描述</td>
        <td width="89%"><input name="Title" type="text" class="input"
            id="Title" size="40" value="" /></td>
    </tr>
    <tr class=line>
        <td width="11%">所属栏目</td>
        <td width="89%"><select name="Category">
            <option value="1">汽车频道</option>

        </select></td>
    </tr>
    <tr class=line>
        <td valign="top">列表页面地址</td>
        <td><input name="EnterUrl" type="text" class="input"
            value="<?php
            echo $row ['EnterUrl']?>" size="60" /></td>
    </tr>
    <tr class=line>
        <td valign="top">列表页匹配描述</td>
        <td><textarea name="ListPreg" cols="80" rows="4" class="input">
        <?php
        echo $row ['ListPreg']?></textarea></td>
    </tr>
    <tr class=line>
        <td valign="top">内容页匹配描述</td>
        <td><textarea name="ContentPreg" cols="80" rows="4" class="input">
        <?php
        echo $row ['ContentPreg']?></textarea></td>
    </tr>
    <tr class=line>
        <td colspan="2" align="center"><input name="Submit" type="submit"
            class="btn" value="提交" /> <input name="Submit" type="button"
            class="btn" value="返回" onclick="history.go(-1);" /></td>
    </tr>
    </form>
</table>
```

【代码解读】

梳理以上代码的执行流程，步骤如下：

（1）在第一次运行 manage.php 程序时，从数据库中提取 spiders 表中的规则信息。

（2）以数组的形式，传送值给表单中对应的标签显示出来。

（3）当对表单中的信息做修改之后，再将表单中的信息以 POST 的方式发送给 PHP 程序处理，然后更新数据库中的内容。

注意： 设置一个隐藏的<input type="hidden" name="flag" value="1">标签，其中标签的值表示程序执行流程的编号，通过判断 flag 标签的值来实现流程控制，请读者在预读代码的时候注意。

运行 manage.php 程序，预览效果如图 8.3 所示，从图中可以看到数据库的规则信息被传到表单中了。

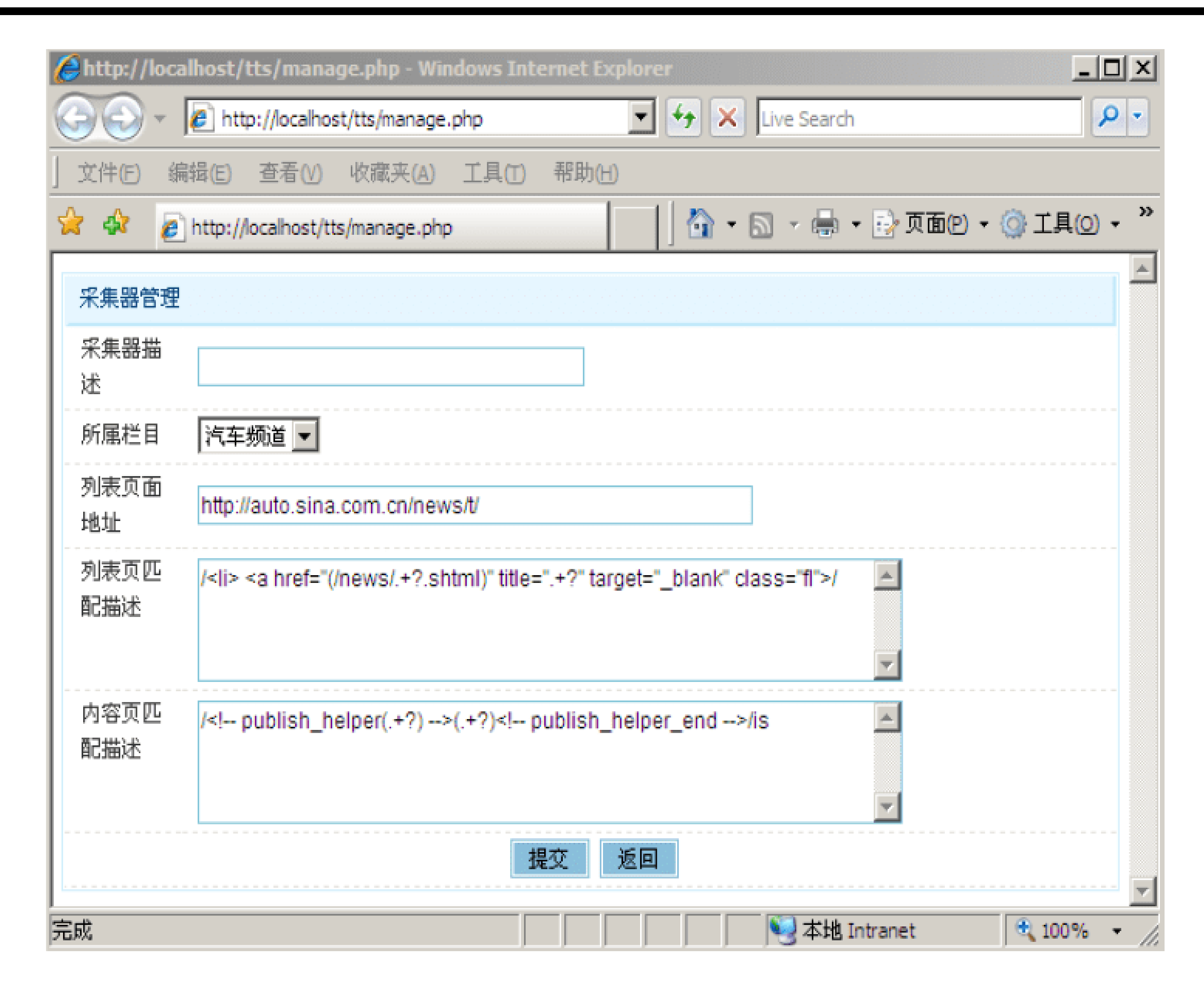


图 8.3　采集器规则修改

8.4.2　爬虫程序

爬虫程序即信息采集程序（spiders.php），通过前几节介绍，读者应该已经对爬虫程序的实现原理有了直观的了解，同时也有了一个爬虫程序的基本模型。下面继续完善这个框架程序的业务流程（如添加信息入库等部分内容）。在完善该程序之前，需要给采集器一个启动的入口，这里设定当采集器规则定制完成后就可以开始采集信息了。打开 manage.php 这个文件找到如下代码：

```
<td colspan="2" align="center"><input name="Submit" type="submit"
    class="btn" value="提交" /> <input name="Submit" type="button"
    class="btn" value="返回" onclick="history.go(-1);" />
</td>
```

将上述代码修改为：

```
<td colspan="2" align="center"><input name="Submit" type="submit"
    class="btn" value="规则提交" /> <input name="Submit" type="button"
    class="btn" value="开始采集" onclick="loadS1()">
</td>
```

【代码解读】

可以看到，当单击“开始采集”按钮后将触发 loadS1()这个 JavaScript 函数，它的功能就是运营采集程序 spiders.php。

下面介绍具体的函数。添加 loadS1()到 manage.php 文件中，loadS1()的代码如下：

```
function loadS1()
{
 window.open ('spiders.php?cid=1')
}
```

其中，cid=1 的意思是加载栏目 id 为 1 的规则开始采集。下面来看具体的 spiders.php

程序的代码实现：

```
<?php
//============================
// 说明：网页采集器程序
//============================
//获得栏目 id
$Category = $ GET ['cid'];
if ($Category) {
    $con = mysql_connect ( 'localhost', 'root', '198251' ) or die ( 'Could
    not connect: ' . mysql error () );
    mysql query ( "set names gb2312" );
    //echo 'Connected successfully';
    $db = mysql select db ( 'get content', $con );
    if (! $db) {
        die ( "Can\'t use download : " . mysql error () );
    } else {
        //获得提取规则数组
        $sql = "SELECT * FROM 'spiders' WHERE 'Category'='" . $Category . "' ";
        $result = mysql_query ( $sql, $con );
        $row = mysql fetch row ( $result );
        //var dump ($row);
        if (! $result) {
            mysql free result ( $result );          //释放结果集
        }
    }
} else {
    exit ( "出错了:(" );
}
$list url = $row [5];//待获取页面的地址
$list_content = file_get_contents ( $list_url );   //获取链接列表
//var dump ($list content);
//观察链接特征：<li> <a href="/news/2010-05-21/1705605024 4.shtml" title="
自主的胜利？" target=" blank" class="fl">自主的胜利？</a><i>
$list_match = $row [2];//获取列表正则规则
$content_match = $row [3];//获得文章内容正则规则
preg match all ( $list match, $list content, $list data );
$i = 0;
//获取到了一个链接列表
//print r($list data[1]);
foreach ( $list data [1] as $detail url ) {
    //处理一下，如果已经抓取过了，就跳过去
    $detail content  =  file get contents  (  "http://auto.sina.com.cn"  .
    $detail url );
    //echo ($i++);
    //获取标题
   preg match ( '/<h1 id="artibodyTitle".+?>(.+?)<\/h1>/U', $detail content,
    $title data );
    $title = $title data [1];
    //die($title);
    //获取内容，需要根据不同特点写正则
    preg match all ( $content match, $detail content, $body data );
    //die($body data[2][0]);
    $body = $body_data [2] [0];
    $sql = "INSERT INTO 'get content'.'articles' ('ID' ,'Title' ,'Click' ,
    'Content' ,'Date' ,'Category' )VALUES (NULL , '" . $title . "', '', '" .
    $body . "', now(), '1');";
    $result = mysql query ( $sql );
}
```

```
mysql_free_result ( $result );//关闭连接
mysql_close ( $db );
?>
```

【代码解读】

本章的爬虫程序的执行流程步骤如下：

（1）根据栏目 id 从数据库中读取对应的规则信息；

（2）通过 file_get_contents()函数获得目标页面的内容；

（3）通过 preg_match_all()函数获取链接列表，然后循环遍历该列表将所需的信息过滤后归档，存入数据的 articles 表中。

8.4.3　获得需采集内容列表

前面介绍的爬虫程序，通过预先设置的规则将数据从目标站点下载到数据库中。现在需要写一个脚本程序将数据库中的内容以列表的形式展示出来，同时还要提供“修改”链接用来编辑、调整内容。采集内容列表程序的代码如下：

```
<?php
$con = mysql_connect ( 'localhost', 'root', 'password' ) or die ( 'Could
not connect: ' . mysql_error () );
mysql_query ( "set names gb2312" );                    //设定数据字符集
echo 'Connected successfully';
$db = mysql_select_db ( 'get_content', $con );         //实例化数据库连接
if (! $db) {
    die ( "Can\'t use download : " . mysql_error () ); //数据库异常报错抛出
} else {
    //获得文件详细信息
    $sql = "SELECT * FROM 'articles' LIMIT 0,10 ";
    $result = mysql_query ( $sql, $con );
    $num_rows = mysql_num_rows ( $result );

    //$rows=mysql_fetch_array($result);
//var_dump ($rows);
}

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gbk" />
</head>
<link href="admin.css" rel="stylesheet" type="text/css" />

<base target="mainFrame">
<div class="m"></div>
<div class="t">
<table align=center cellspacing=0 cellpadding=0>
    <tr class=head>
        <td colspan="4">栏目名称</td>
    </tr>
    <tr class=line>
        <td><strong>日期</strong></td>
        <td width="913"><strong>标题</strong></td>
```

```
        <td><strong>修改</strong></td>

    </tr>
<?php
for($i = 0; $i < $num rows; $i ++) {
?>
<tr class=line>
        <td width="135">
<?php
    $row = mysql fetch array ( $result );
    echo $row ['Date'];
?>
</td>
        <td>
<?php
    echo $row ['Title'];
?>
</td>
        <td width="83"><a
            href="articles save.php?aid=<? phpecho $row ['ID'];?>" target=" top">
            修改</a></td>
</tr>
<?php
}
?>
<tr class=line>
        <td colspan="4" align="center"> </td>
    </tr>
</table>
</div>
```

将上述代码保存为articles.php 文件。上述代码并不复杂，核心内容是将数据库中articles 表中的内容取出，以列表形式输出到浏览器中。在浏览器中运行 articles.php 文件的效果如图 8.4 所示。

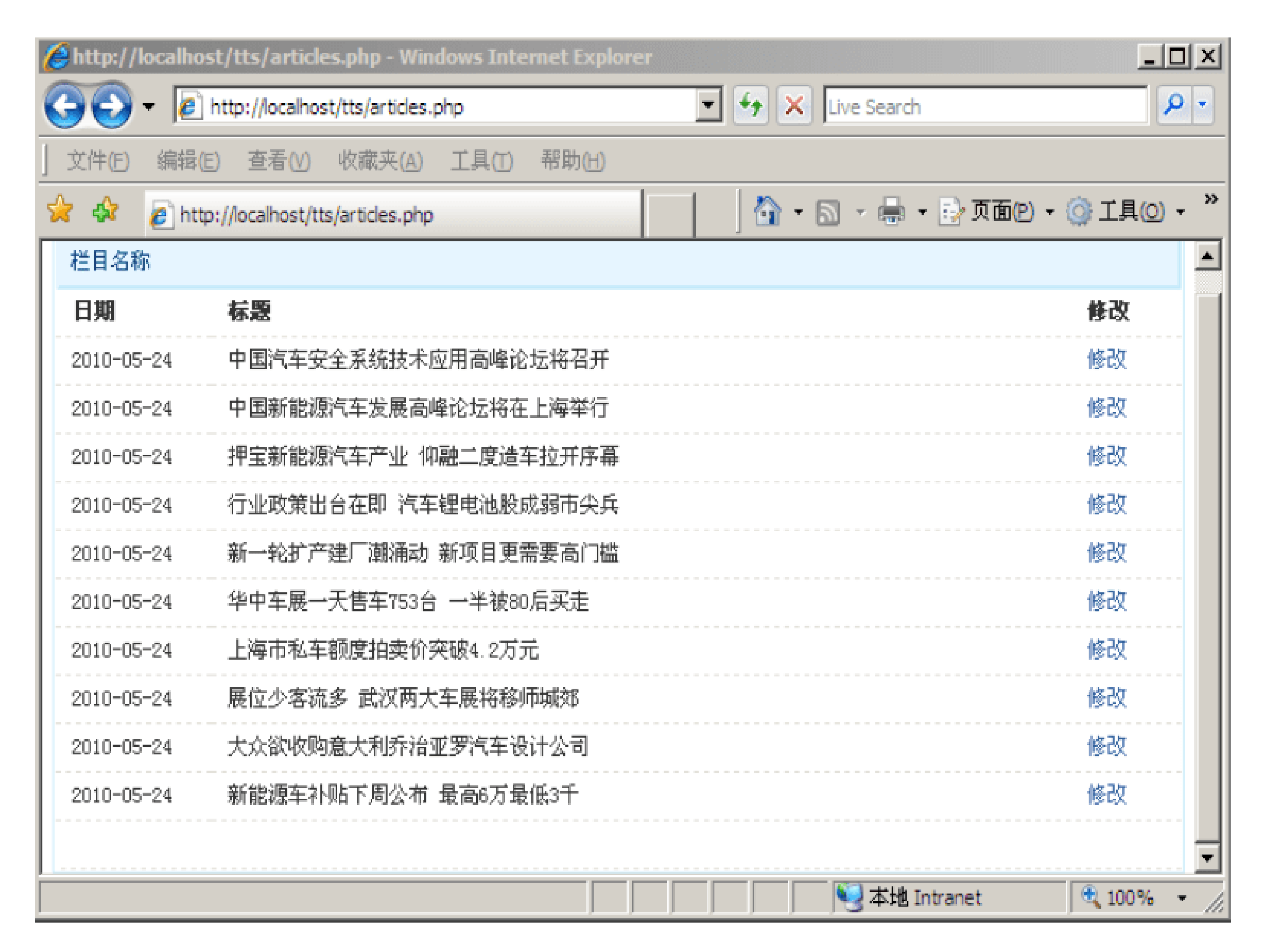


图 8.4　文章列表

8.4.4　内容编辑器

内容编辑器被广泛地应用在各种信息管理后台，它的作用是在线查看和编辑 HTML 代码和脚本程序。在本章中使用内容编辑器（FCKeditor）对文章内容进行预览和编辑。

FCKeditor 是一个专门使用在网页上的文字编辑器，属于开放源代码的所见即所得形式，它是轻量级、不需要复杂安装过程的文字编辑器。下面来了解 FCKeditor 在 PHP 中的配置方法及使用。

1. 下载及解压

从下面的地址下载新版本 FCKeditor：

```
http://www.fckeditor.net/download
```

将下载后的文件解压到 Apache 的 Web 发布目录 www/中。

2. 调用FCKeditor

新建一个 PHP 文件，并在程序中加载 FCKeditor 文件 include("FCKeditor/fckeditor.php")，然后加入初始化 FCKeditord 的各项配置。初始化及配置选项代码如下：

```
<?php
//功能：FCK 初始化及配置
$oFCKeditor = new FCKeditor ( 'FCKeditor1' );       //建立对象
$oFCKeditor->BasePath = 'FCKeditor/';               //FCKeditor 所在的位置
$oFCKeditor->ToolbarSet = 'Default';                //工具按钮
$oFCKeditor->Create ( 'EditorDefault', '60%', 150 );
?>
```

【代码解读】

以上的代码实例化编辑器类，并设置了 FCKeditor 编辑器的基本参数。通过 $oFCKeditor->Value 的方法，将文章内容部分输出到编辑器中。当编辑器中的内容改变后，再将内容变量更新到对应的数据库字段中，完成文本预览和编辑。完成上述功能的完整代码如下：

```
<?php
$aid = $_GET ['aid'];                               //文章索引 id
if ($aid) {
    $con = mysql_connect ( 'localhost', 'root', '198251' ) or die ( 'Could
    not connect: ' . mysql_error () );
    mysql_query ( "set names gb2312" );             //设定数据字符集
    //echo 'Connected successfully';
    $db = mysql_select_db ( 'get_content', $con ); //实例化数据库连接
    if (! $db) {
        die ( "Can\'t use download : " . mysql_error () );
                                                    //数据库链接异常报错抛出
    } else {
        //获得文件详细信息
        $sql = "SELECT * FROM 'articles' WHERE 'ID' ='" . $aid . "' LIMIT 1 ";
        $result = mysql_query ( $sql, $con );
        $row = mysql_fetch_array ( $result );       //获得内容结果集
```

```
    }
}

$fck = $_POST ["FCKeditor1"];                         //获得修改后的内容变量
if ($fck != "") {
    $sql1 = "UPDATE 'get_content'.'articles' SET 'Content' = '" . $fck . "'
    WHERE 'articles'.'ID' ='" . $aid . "' LIMIT  1; ";
    echo $sql1;
    $result = mysql_query ( $sql1, $con );
    if ($result) {
        mysql_free_result ( $result );                //释放结果集
    header("location:list.php");                      //重定向浏览器
    }
}
?>
<!--页面模板部分-->
<table align=center cellspacing=0 cellpadding=0>
    <form action="articles_save.php?aid=<?php
    echo $aid;
    ?>" method="post">
    <tr class=head>
        <td colspan="2"><?php
        echo $row ['Title']?></td>
    </tr>
    <tr class=line>
        <td width="9%">标题</td>
        <td width="91%"><input name="Title" type="text" class="input"
            id="Title" size="70" value="<?php
            echo $row ['Title']?>" /></td>
    </tr>
    <tr class=line>
        <td valign="top">内容</td>
        <td>
<?php
include ("FCKeditor/fckeditor.php");                  //加载入口文件
$oFCKeditor = new FCKeditor ( 'FCKeditor1' );         //建立对象
$oFCKeditor->BasePath = 'FCKeditor/';                 //FCKeditor 所在的位置
$oFCKeditor->ToolbarSet = 'Default';                  //工具按钮
$oFCKeditor->Value = $row ['Content'];
$oFCKeditor->Create ( 'EditorDefault', '60%', 150 );    //创建工作区域
?>
</td>
    </tr>
    <tr class=line>
        <td colspan="2" align="center"><input name="Submit" type="submit"
            class="btn" value="提交" /> <input name="Submit" type="button"
            class="btn" value="返回" onclick="history.go(-1);" /></td>
    </tr>
    </form>
</table>
</div>
```

注意：观察以上的代码，这段代码是通过程序获得$_GET[ID]的值（即文章的索引 id）将对应文章的内容部分从数据库中取出提交到编辑器中，编辑内容的保存也是同样的原理。运行上面的代码，效果如图 8.5 所示。

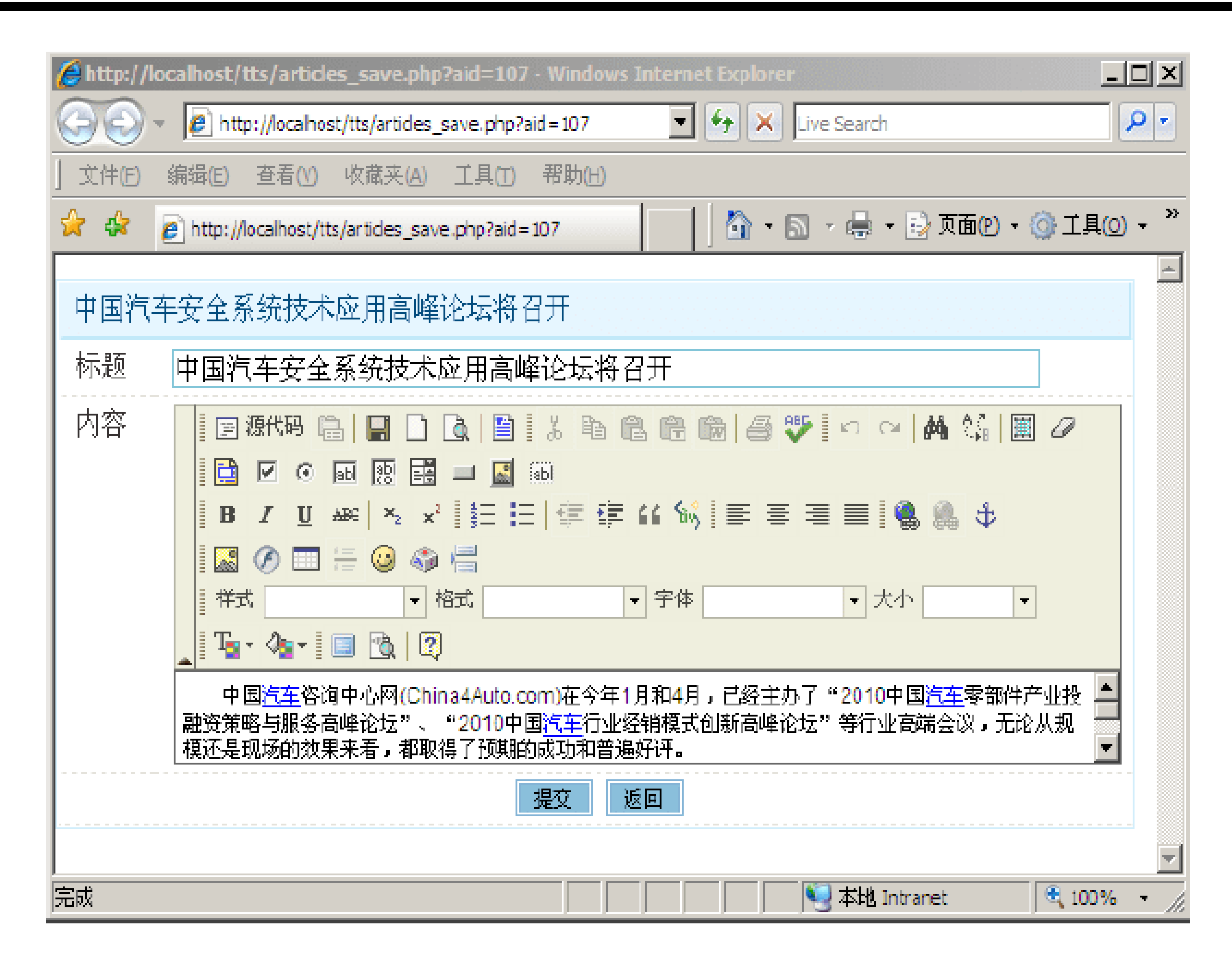


图 8.5　FCKeditor 内容编辑器

8.5　采集异常及处理

前几节介绍了内容自动采集器模块的基本原理，并完成了采集器的编码。现在采集器程序已经可以从目标站点抓取需要的内容了，不过当被抓取站点的内容过多时，就会引发一系列的问题导致程序崩溃，内容抓取失败。本节将介绍采集过程中常见的异常状况，并说明这些状况产生的原因和处理方法。

8.5.1　抓取超时解决方法

众所周知，PHP 通过浏览器执行的时间限制是 30 秒，超过这个时间限制程序会被判定异常，系统会报出一个“致命级”的错误终止程序运行。但是要想全面理解这个问题并找到解决办法，需要理解 PHP 的“连接处理”机制。在 PHP 内部系统维护着连接状态，其状态有 3 种可能的情况：

- 0 - NORMAL（正常）；
- 1 - ABORTED（异常退出）；
- 2 - TIMEOUT（超时）；

当连接时间超过 PHP 的时限（请参阅 set_time_limit()函数）时，TIMEOUT 状态的标记将被打开。该标记决定脚本是否需要在客户端中断连接时退出。有时让脚本完整地运行会带来很多方便，即使没有远程浏览器接受脚本的输出。

默认的情况是，当远程客户端连接中断时脚本将会退出。该处理过程可由 php.ini 的 ignore_user_abort 或由 Apache .conf 设置中对应的 php_value ignore_user_abort，以及 ignore_user_abort()函数来控制。如果没有“告诉”PHP 忽略用户的中断，脚本将会被中断，除非通过 register_shutdown_function()设置了关闭触发函数。通过该关闭触发函数，当远程用户单击 STOP 按钮后，脚本再次尝试输出数据时，PHP 将会检测到连接已被中断，并调用关闭触发函数，但是脚本也有可能被内置的脚本计时器中断。

🔔注意：默认的超时限制为 30 秒。这个值可以通过设置 php.ini 的 max_execution_time 或 Apache .conf 设置中对应的 php_value max_execution_time 参数或者 set_time_limit() 函数来更改。

通过上面的分析，可以了解采集程序抓取内容超时的原因。下面为读者介绍两种常用的解决方法：

第一种思路是通过延时函数，代码如下：

```
$ch = curl init();
$timeout = 10;
curl setopt ($ch, CURLOPT URL, $url);
curl setopt ($ch, CURLOPT RETURNTRANSFER, 1);
curl setopt ($ch, CURLOPT CONNECTTIMEOUT, $timeout);
$handles = curl exec($ch);
curl_close($ch);
```

将上面的代码加入到脚本代码中，就可以解决执行超时的问题了。

第二种思路是通过命令行的方式执行 PHP 程序，这样就可以有效的避免 30 秒限制所产生的问题了。

8.5.2　任务接力模式

任务接力模式的本质是对一个任务做拆分，将一个任务拆分成多个子任务来实现。如执行完付费任务时，使用 PHP 的 Header()函数重定向到发送 Email 的页面，完成后再定向到其他页面，采用 redirect 页面的方式来接力地完成任务。

设计 3 个 PHP 文件，文件名分别是 s1.php、s2.php、s3.php，让这 3 个文件进行程序接力。S1.php 文件的内容如下：

```
<?php
ob start ();
sleep ( 3 ); //代表做某些任务所用的时间
echo "在做第一件事。";
header ( "Location: http://127.0.0.1/s3.php?i=1" );
ob end flush ();
?>
```

S2.php 文件的内容如下：

```
<?php
ob start ();
$i = $ GET ['i'] + 1;
echo "在做第二件事"
;
sleep ( 3 ); //用来代表做了某件事件
header ( "Location: http://127.0.0.1/s4.php?i=$i" );
ob_end_flush ();
?>
```

S3.php 文件的内容如下：

```
<?php
ob_start ();
$i = $_GET ['i'] + 1;
```

```
sleep ( 3 );
echo "在做第三件事";
ob end flush ();
?>
```

【代码解读】

上述 3 个文件是将一个任务拆分成 3 个子任务来接力执行，具体是使用 PHP 的 Header() 函数重定向到当前的执行程序，完成后再定向到程序继续执行。当其他程序执行时，当前程序通过 sleep()函数暂停执行，这样各个程序就可以通过接力的方式完成任务。

8.6　小　　结

本章的重点内容是介绍内容采集的基本原理和实现方法，使读者可以直观地了解正则表达式在内容采集过程中的核心作用。本章首先介绍内容采集模块的技术原理，在第 8.1 节介绍了内容自动采集器功能及采集原理。在第 8.2 节结合具体的需求设计了两个存储表（规则存储表和采集内容表）。第 8.3 节为读者介绍了采集器模块界面设计，本节包含采集器管理界面和采集内容列表界面。第 8.4 节是具体的采集器爬虫程序实现，本节包含两个组成部分，即采集规则管理程序、爬虫程序。最后将上述两个程序整合到获得需采集内容列表页面中，然后介绍了内容编辑器的设置及使用。在第 8.5 节介绍了采集过程中的异常及处理办法。

在学习本章时读者需要掌握程序流程控制、PHP 中的正则表达式实现、FCKeditor 内容编辑器、抓取程序异常分析及处理、任务接力模式，以上内容是采集器程序的核心技术要点。

第 9 章　日程表模块（PHP+AJAX+XML）

日程表模块程序是典型的 PHP 与 AJAX 交互的应用，这类程序应用非常广泛，如一些培训网站会按日期发布课程通知、学员根据自己的时间安排选择不同的课程及上课时间。这是一个很实用的功能模块，可以被改造成不同的应用，如个人日程表等。本章通过课程表模块的实现将 AJAX 技术介绍给读者，读者可以通过本章了解到 AJAX 的基本原理、PHP 与 AJAX 是如何交互的，以及通过数据库驱动的 AJAX。

本章主要涉及的知识点如下。

- AJAX 含义：AJAX（Asynchronous JavaScript and XML）是指一种创建交互式网页应用的网页开发技术。需要注意的是，AJAX 不是一种新的编程语言，而是一种使用现有标准的新方法，是多种技术的组合。
- XMLHttpRequest 对象的属性：该对象包含 3 个基本属性，分别是 onreadystatechange 属性、readyState 属性、responseText 属性。
- AJAX 中数据传输：AJAX 应用程序是通过 XML 格式文档做数据载体的。
- PHP 时间函数：PHP 提供 date()函数用来处理日期相关字符串得到标准时间格式。

9.1　AJAX 基础

AJAX（Asynchronous JavaScript and XML）是指一种创建交互式网页应用的网页开发技术。需要注意的是，AJAX 不是一种新的编程语言，而是一种使用现有标准的新方法，它是多种技术的组合。通过 AJAX 可以创建更好、更快、更友好的 Web 应用程序，它使用 XHTML 和 CSS 标准化呈现，使用 DOM 实现动态显示和交互，使用 XML 和 XSTL 进行数据交换与处理，使用 XMLHttpRequest 对象进行异步数据读取，使用 JavaScript 绑定和处理所有数据。更重要的是 AJAX 打破了使用页面重载的惯例技术组合。

9.1.1　HTTP 请求和响应基础

在常见的 JavaScript 编程中，如果需要从服务器中的文件或数据库中得到任何的信息，或者向服务器发送信息的话，就必须通过 HTML 表单使用 GET 或 POST 的方法，向服务器提交数据。传统的 Web 应用程序结构如图 9.1 所示，当用户单击“提交”按钮后等待服务器的响应，然后页面会加载返回结果。由于每当用户提交输入后，服务器都会将全部页面的元素返回页面，这种方式实际造成了严重的资源浪费。相同的部分没有被重复的使用，哪怕是最微小的改变都要加载全部的页面元素，这种重载方法使得 Web 应用变得越来越不友好。

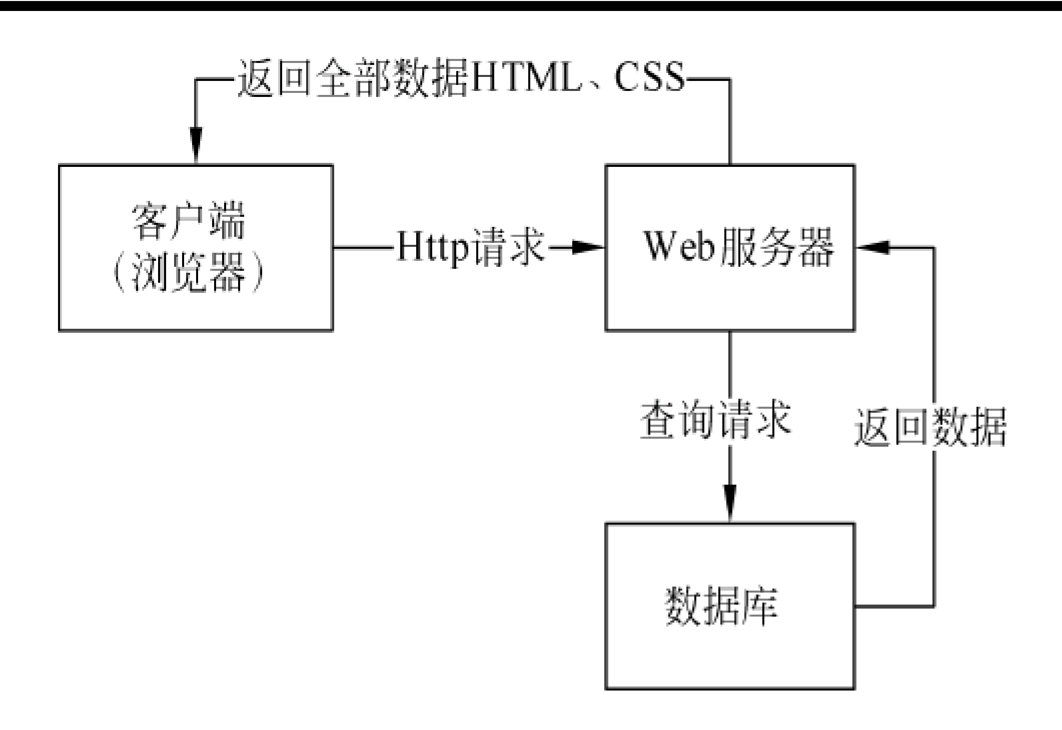


图 9.1　传统 Web 应用程序结构

在 AJAX 模式中，JavaScript 通过 JavaScript 的 XMLHttpRequest 对象，直接与服务器来通信，AJAX 模式的应用程序结构如图 9.2 所示。

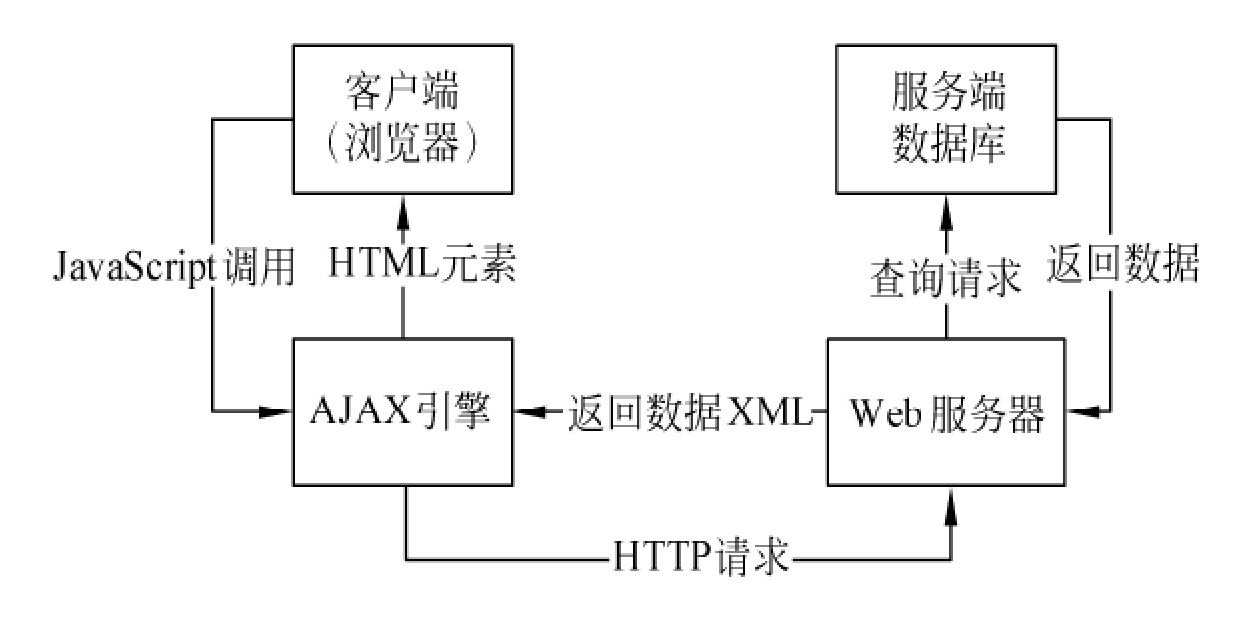


图 9.2　AJAX 模式应用程序结构

通过上图描述的 AJAX 执行流程，可以总结出一个 AJAX 应用程序与服务器进行一次交互的过程，分为以下几个步骤。

（1）用户在 Web 页面上执行了某个操作，如单击某个链接或选中某个选项等。

（2）根据用户的操作，页面产生相应的 DHTML 事件。

（3）调用注册到该 DHTML 事件的客户端 JavaScript 事件处理函数。其中，需要创建并初始化一个用以向服务器发送异步请求的 XMLHttpRequest 对象，同时指定一个回调函数。当服务器端的响应返回时，将自动调用该回调函数。

（4）服务器收到 XMLHttpRequest 对象的请求之后，根据请求进行一系列的处理。

（5）处理完毕，服务器端向客户端返回所需要的数据。

（6）数据到达客户端之后执行 JavaScript 回调函数，并根据返回的数据对用户显示界面进行更新。

（7）用户获得自己操作所需的数据，对应看到显示界面的变化（数据更新）。

通过使用 HTTP 请求，Web 页面可向服务器进行请求，得到来自服务器的响应，并在客户端采用 JavaScript 处理来自服务器的响应。因为服务器和浏览器之间交换的数据大量减少，从而使得 Web 应用响应更快。同时，很多处理工作可以在发出请求的客户端机器上完成，Web 服务器的处理量减少了。由于是异步更新整个过程没有页面的刷新，所以具有很高的用户交互性，几乎觉察不到延迟的 Web 应用。

总地来说，AJAX 模式相比传统 Web 应用程序具有以下特点：

- 通过异步模式，提升用户体验。

- 优化浏览器和服务器之间的数据传输，减少不必要的往返数据，减少了带宽占用量。
- AJAX 引擎在客户端运行，承担了一部分原本由服务器承担的工作，从而减少了大用户量下的服务器负载。

9.1.2　XMLHttpRequest 对象

为了用 JavaScript 向服务器发送一个 HTTP 请求，需要一个具备这种功能的类实例，这样的类由 Internet Explorer 以 ActiveX 对象引入，被称为 XMLHTTP。在向服务器发送数据之前，先来了解 XMLHttpRequest 对象的 3 个重要的属性。

1．onreadystatechange属性

onreadystatechange 属性存有处理服务器响应的函数。下面的代码定义一个空的函数，可同时对 onreadystatechange 属性进行设置：

```
xmlHttp.onreadystatechange=function()
{
//逻辑代码部分
}
```

2．readyState属性

readyState 属性储存服务器响应的状态信息。当 readyState 改变时，onreadystatechange 函数就会被执行。readyState 属性返回的状态值如表 9.1 所示。

表 9.1　readyState属性返回的状态值

状态	描　述
0	请求未初始化（在调用 open()之前）
1	请求已提出（调用 send()之前）
2	请求已发送（这里通常可以从响应得到内容头部）
3	请求处理中（响应中通常有部分数据可用，但是服务器还没有完成响应）
4	请求已完成（可以访问服务器响应并使用它）

下面向这个 onreadystatechange 函数添加一条判断语句，用来测试是否获得服务器响应。代码如下：

```
xmlHttp.onreadystatechange=function()
{
    if (xmlHttp . readyState == 4) {
        // 从服务器的 response 获得数据
    }
}
```

3．responseText属性

针对纯文本格式及 XML 格式的响应数据，XMLHttpRequest 对象提供了对应的两个属性，一个是 responseText 属性，即将响应提供为一个串；另一个是 responseXML 属性，即将响应提供为一个 XML 对象，可以通过 responseText 属性来取回由服务器返回的数据。

下面的示例代码中，将把文本框的值设置为 responseText 的值，代码如下：

```
xmlHttp.onreadystatechange=function()
{
    if (xmlHttp . readyState == 4) {
    document.myForm.test.value=
        xmlHttp . responseText;
    }
}
```

9.1.3　向服务器发送请求

当将请求发送到服务器时，需要使用 open()方法和 send()方法。open()方法需要 3 个参数，第一个参数定义发送请求所使用的方法（是 GET 方法还是 POST 方法）；第二个参数规定服务器端脚本的 URL；第三个参数规定应当对请求进行异步地处理。代码形式如下：

```
var url;
//alert (url);
if(window.XMLHttpRequest) {          //判断是否已经加载对象
    req = new XMLHttpRequest();      //实例化对象
} else if(window.ActiveXObject) {
    req = new ActiveXObject("Microsoft.XMLHTTP");
}
req.open("GET", url, true);
```

send()方法可将请求送往服务器，这里需要注意，URL 即服务端处理程序的路径问题。代码如下：

```
var url;
//alert (url);
if(window.XMLHttpRequest) {          //判断是否已经加载对象
    req = new XMLHttpRequest();      //实例化对象
} else if(window.ActiveXObject) {
    req = new ActiveXObject("Microsoft.XMLHTTP");
}
req.open("GET", url, true);
req.send(null);
```

下面是一个严谨的 AJAX 引擎函数。其中，对各种异常情况作了必要的处理。具体代码如下：

```
<script type="text/JavaScript">
//AJAX 引擎函数
function AJAXFunction()
{
 var req;
 //服务端处理程序
 var url ;
 //判断浏览器类型
 try
 {
  //Firefox, Opera 8.0+, Safari; 浏览器
  req=new reqRequest();
  }
 catch (e)
 {
```

```
 //Internet Explorer；浏览器
 try
 {
  req=new ActiveXObject("Msxml2.req");          //注册浏览器对象
  }
  catch (e)
  {
    try
    {
     req=new ActiveXObject("Microsoft.req");   //注册浏览器对象
     }
     catch (e)
     {
        alert("您的浏览器不支持 AJAX! ");           //弹出错误提示
        return false;
      }
      }
    }
    req.onreadystatechange=function()           //onreadystatechange 状态方法
      {
      if(req.readyState==4)
        {
         document.myForm.text.value=req.responseText;
        }
      }
    req.open("GET",url,true);                   //获得 url
    req.send(null);                             //发送请求
 }
</script>
```

当然，只有 AJAX 引擎函数还是不够的，需要页面的触发机制才能执行 AJAX。比如，通过 onClick 单击触发事件、onmouseover 和 onmouseout 鼠标移动触发事件等。

9.2　AJAX 中的服务端脚本

前一节介绍了 AJAX 的基础知识，它并不是一种新的编程语言，而仅仅是一种新的技术，它可以创建更好、更快且交互性更强的 Web 应用程序。AJAX 使用 JavaScript 在 Web 浏览器与 Web 服务器之间发送和接收数据。

通过异步的与 Web 服务器交换数据，而不是每当用户“浏览”时就重载整个 Web 页面，AJAX 技术可以使网页更迅速地响应。很多脚本语言都可以用来做 AJAX 的服务端脚本，在实现原理上都是大同小异。本节将介绍使用 PHP 来处理客户端提交过的异步的数据请求。

9.2.1　AJAX 与 XML 文件进行交互式通信

现在来回顾下 AJAX 模式实现的原理。首先在客户端创建一个 XMLHttpRequest 对象，这是 AJAX 的核心，用来与服务器进行异步通信。然后为该对象的 onreadystatechange 属性添加一个事件，当对象的 readyState 改变时就会触发指定的事件，这时发送请求读取服务器端的 XML 数据了。最后利用 getElementsByTagName 解析 responseXML 返回的值，完成数据的最后处理。

通过上面的执行可以看出，XML 是 AJAX 数据传输的核心，那么什么是 XML 呢？下面从不同角度给出相应的解释。

- XML 指可扩展标记语言（EXtensible Markup Language）。
- XML 是一种标记语言，类似 HTML。
- XML 的设计宗旨是传输数据，而非显示数据。
- XML 标签没有被预定义，需要自行定义标签。
- XML 被设计为具有自我描述性。
- XML 是 W3C 的推荐标准。

XML 被设计为传输和存储数据，其核心是存储数据信息。它不同于 HTML 被设计用来显示数据，其焦点是数据的外观。可以更简要的描述为 HTML 旨在显示信息，而 XML 旨在传输信息。目前 XML 的应用已经无所不在，已经被作为各种应用程序之间进行数据传输的最常用的工具，并且在信息存储和描述领域变得越来越流行。

下面通过一个例子，来直观地了解 AJAX 是如何与 XML 文件进行交互式通信的。

（1）创建一个标准的 XML 文档用来存储数据。文件命名为 Ajax.xml 内容如下：

```
<?xml version="1.0" encoding="utf-8" ?>
<family>
<member>
   <name>Mark</name>
   <doing>task1</doing>
</member>
<member>
   <name>Annie</name>
   <doing>task2</doing>
</member>
<member>
   <name>Nathan</name>
   <doing>task3</doing>
</member>
<member>
   <name>Rob</name>
   <doing>task4</doing>
</member>
<member>
   <name>Chris</name>
   <doing>task5</doing>
</member>
</family>
```

（2）创建一个文件名为 AjaxDemo.html 的文件，来读取和处理 XML 文档的数据，获得数据后再通过一个 2 列 5 行的表格显示出来。具体的代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Ajax-xml</title>
<script type="text/JavaScript">
<!--
function GetXmlHttpObject(){
xmlHttp=null;
try{
   xmlHttp = new XMLHttpRequest();     //针对 Firefox、Opera 及 Safari 浏览器
}
catch(e){
```

```
   try{
    xmlHttp = new ActiveObject("Msxml2.XMLHTTP");
                                                  //针对 Internet Explorer 6.0+
   }
   catch(e){
    try{
     xmlHttp = new ActiveObject("Microsoft.XMLHTTP");
                                                  //针对 Internet Explorer 5.5+
    }
    catch(e){
     alert('对不起!您的浏览器不支持 AJAX');       //弹出错误提示信息（AJAX 失败）
     return false;
    }
   }
}
return xmlHttp;
}
//改变状态
function state Changed(){
if(xmlHttp.readyState!=4)return;
if(xmlHttp.status!=200){
   alert('加载 XML 文件失败');                   //弹出错误提示信息（XML 失败）
   return;
}
txt="<table border='1'>";
x=xmlHttp.responseXML.documentElement.getElementsByTagName("member");
for (i=0;i<x.length;i++)
   {
   txt=txt + "<tr>";
   xx=x[i].getElementsByTagName("name");      //传回名称为 name 的元素集合
     {
     try
       {
       txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";
       }
     catch (er)
       {
       txt=txt + "<td> </td>";
       }
     }
   xx=x[i].getElementsByTagName("doing");     //传回名称为 doing 的元素集合
     {
     try
       {
       txt=txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";//子节点的值
       }
     catch (er)
       {
       txt=txt + "<td> </td>";
       }
     }
   txt=txt + "</tr>";
   }
txt=txt + "</table>";
document.getElementById('name').innerHTML = txt;        //为 name 元素赋值
}
function showXml(url){
xmlHttp = GetXmlHttpObject();
if(xmlHttp!=null){
   xmlHttp.onreadystatechange = state_Changed;          //修改对象状态
```

```
   xmlHttp.open("GET",url,true);
   xmlHttp.send(null);
}
}
//-->
</script>
</head>
<body>
<p><span id="name"></span></p>
<form><input type="button" value="显示 XML" id="ok" name="ok"
    onclick="showXml('Ajax.xml')" /></form>
</body>
</html>
```

（3）在浏览器中执行 AjaxDemo.html 文件，预览效果如图 9.3 所示。

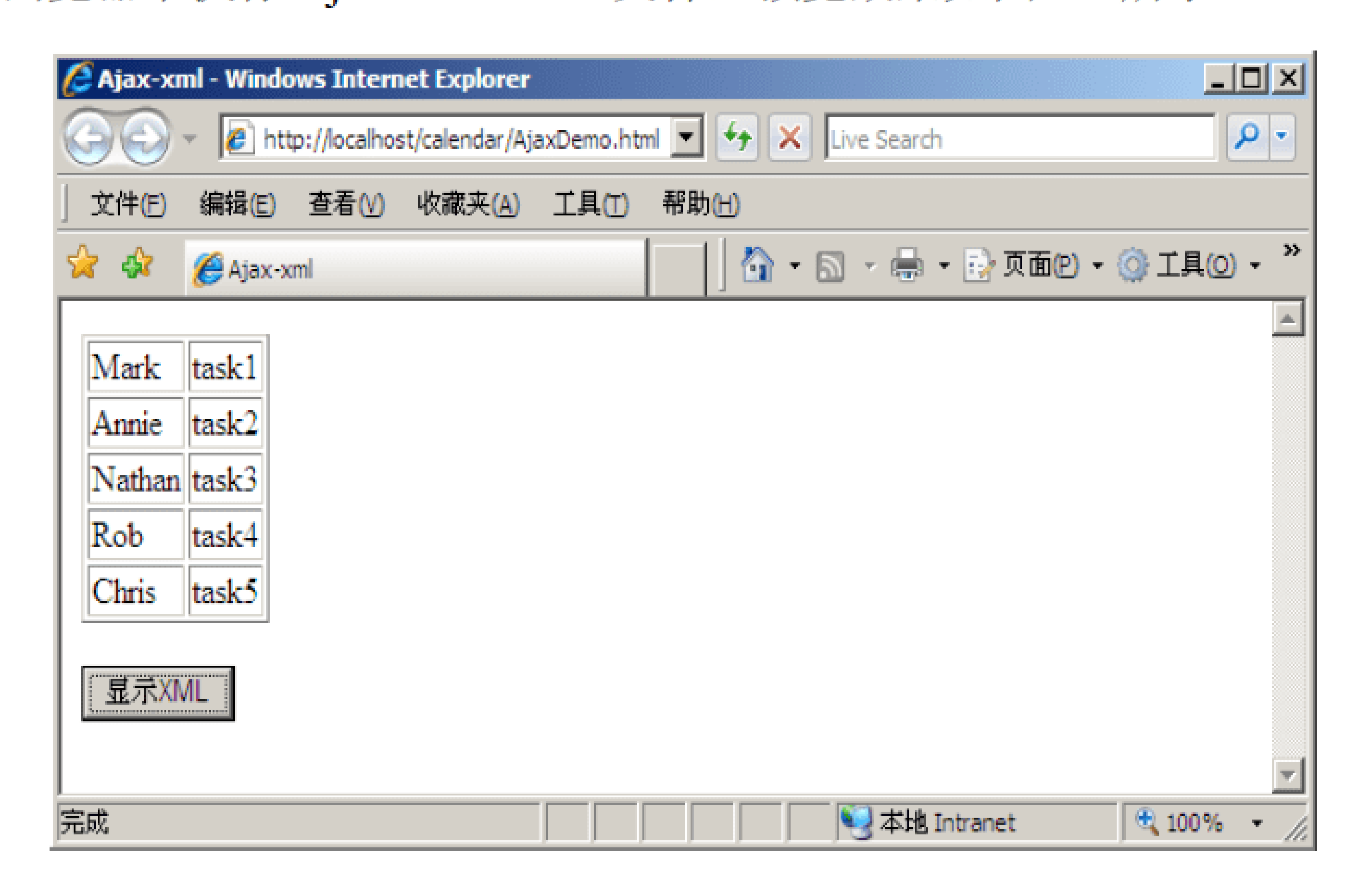


图 9.3　AJAX 与 XML 文件数据通信

9.2.2　PHP 生成 XML

前面介绍了 AJAX 与 XML 交互通信的方式，接下来的问题是 XML 本身只是数据的一个“容器”，并不能动态地获得数据，那么该如何获得数据呢？其实解决方法很简单，PHP 脚本既然可以生成动态 HTML，当然也可以生成动态 XML。

用 PHP 生成 XML 文档并不复杂，只需要用 header()函数把文档的 MIME 类型改成 text/xml 即可。为了避免<?xml ... ?>声明被解释为一个 PHP 标签，需要编辑 php.ini，将 short_open_tag 选项设为不启用，或者直接用 echo 把这一行打印出来：

```
<?php
echo '<?xml version="1.0" encoding="utf-8" ?>';
?>
```

下面的代码将生成一个标准的 XML 格式文件，代码如下：

```
<?php
header ( 'Content-Type: text/xml' );
echo '<?xml version="1.0" encoding="utf-8" ?>';
echo '<family>';
echo '
<member>
   <name>Mark</name>
```

```
    <doing>task1</doing>
</member>
<member>
    <name>Annie</name>
    <doing>task2</doing>
</member>
<member>
    <name>Nathan</name>
    <doing>task3</doing>
</member>
<member>
    <name>Rob</name>
    <doing>task4</doing>
</member>
<member>
    <name>Chris</name>
    <doing>task5</doing>
</member>';
echo '</family>';
?>
```

【代码解读】

上面的代码对读者来说是非常容易理解的，它按照 XML 要求的格式输出给浏览器。这里需要注意的是，由于是使用 header()函数声明文件的类型，所以在 header()函数之前不能有任何其他字符的输出，否则 WebServer 将会报错。另外，在输出 XML 格式时，也要保证不能有其他多余的输出，这样保证浏览器可以正确解析。

9.2.3　数据库驱动的 AJAX

正如前文介绍的 XML 只是数据的载体，因此不适合做数据存储。如果 XML 不能与数据库建立联系那么它的优势将变得毫无用处，毕竟 XML 的丰富与强大是需要有背后的数据库做支撑的。本节将通过一个例子，来展示如何通过数据驱动 AJAX。

（1）首先创建一个数据库脚本，用来存放数据资料，代码如下：

```
SET SQL MODE="NO AUTO VALUE ON ZERO";

--
-- 数据库: 'test1'
--

-- --------------------------------------------------------

--
-- 表的结构 'xml'
--

CREATE TABLE 'xml' (
  'id' int(10) NOT NULL auto_increment,
  'name' varchar(20) NOT NULL,
  'doning' varchar(20) NOT NULL,
  PRIMARY KEY  ('id')
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO INCREMENT=6 ;

--
-- 导出表中的数据 'xml'
--
```

```
INSERT INTO 'xml' VALUES (1, 'Mark', '任务 1');
INSERT INTO 'xml' VALUES (2, 'Annie', '任务 2');
INSERT INTO 'xml' VALUES (3, 'Nathan', '任务 3');
INSERT INTO 'xml' VALUES (4, 'Rob', '任务 4');
INSERT INTO 'xml' VALUES (5, 'Chris', '任务 5');
```

（2）创建一个文件名为 xml.php 的脚本文件，用来读取数据库中的数据，并生成标准的 XML 格式文件。代码如下：

```
<?php
header ( 'Content-Type: text/xml' );          //header()函数声明文件类型
echo '<?xml version="1.0" encoding="utf-8" ?>';
echo '<family>';
$db = mysql_connect ( $dbhost, $dbuser, $dbpass );     //数据库连接信息
mysql_select_db ( 'test1' );                           //选择指定的数据库
$sql = "SELECT * FROM 'xml' LIMIT 0 , 30 ";
$result = mysql_query ( $sql );
$num_rows = mysql_num_rows ( $result );
for($i = 0; $i < $num_rows; $i ++) {
    $array [$i] = mysql_fetch_row ( $result );
    //以下输出标签组成元素
    echo "<member>";
    echo "<name>" . $array [$i] [1] . "</name>";
    echo "<doing>" . $array [$i] [2] . "</doing>";
    echo "</member>";
}
echo '</family>';//输出标签
?>
```

【代码解读】

上述代码中，使用 header()函数给浏览器声明一个表明文件类型的“头”，再通过 echo 函数将标签元素直接输出到浏览器中，完成读取和生成 XML 格式文件。

（3）执行以上代码将获得一个标准的 XML 文档，效果如图 9.4 所示。

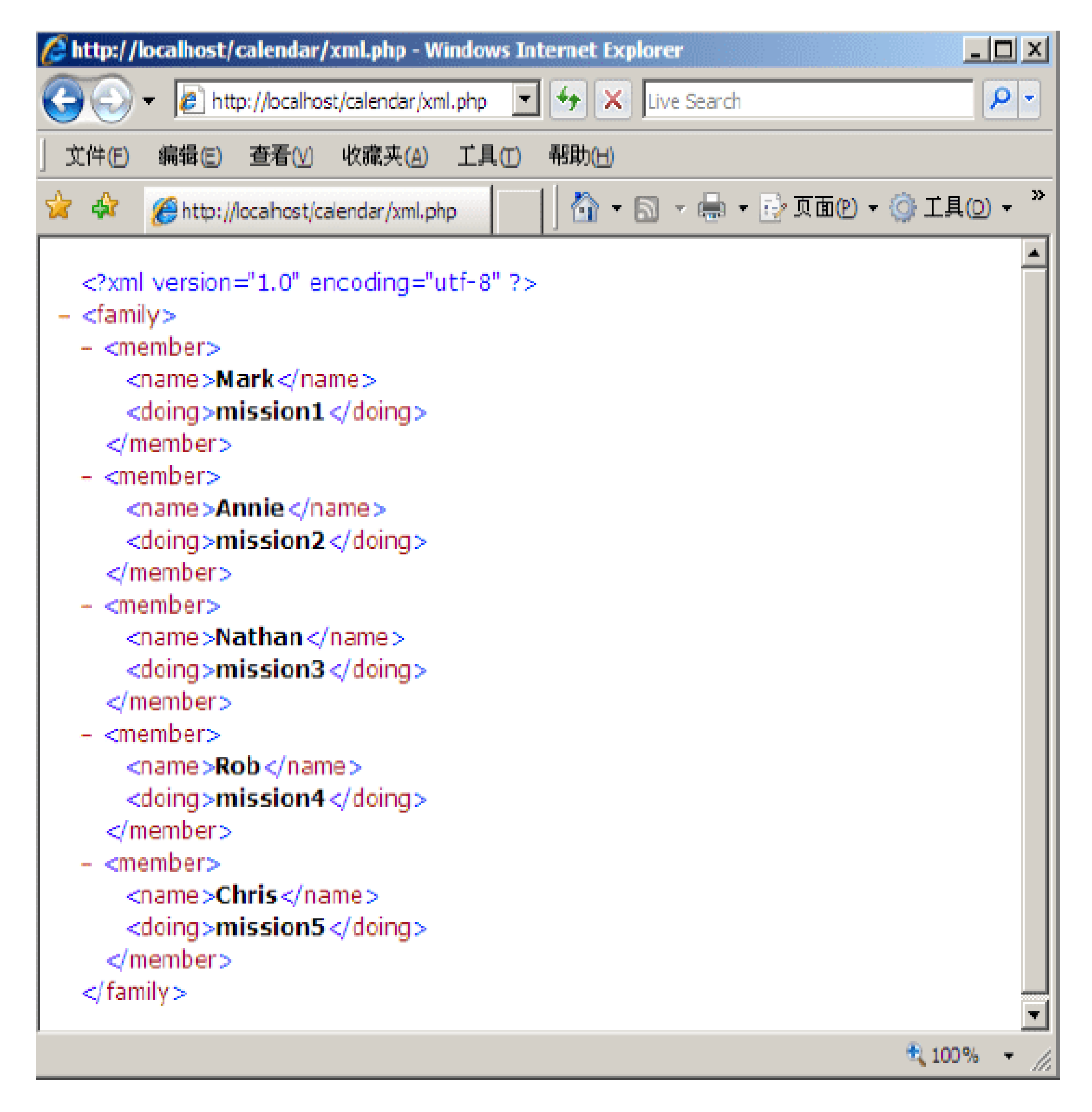


图 9.4　数据库中信息生成 XML 文档

（4）打开 AjaxDemo.html 文件，对页面部分作修改，将获取文件替换为 xml.php，这样就完成了动态的从数据库中获得数据并显示的功能了。修改的部分代码如下：

```
<body>
<p>
    <span id="name"></span>
</p>
<form>
<input type="button" value="显示 XML" id="ok" name="ok"onClick="showXml
('xml.php')" />
</form>
</body>
```

（5）在浏览器中重新运行 AjaxDemo.html 文件，效果如图 9.5 所示，可以看到第二列部分信息变成数据库中的内容了。

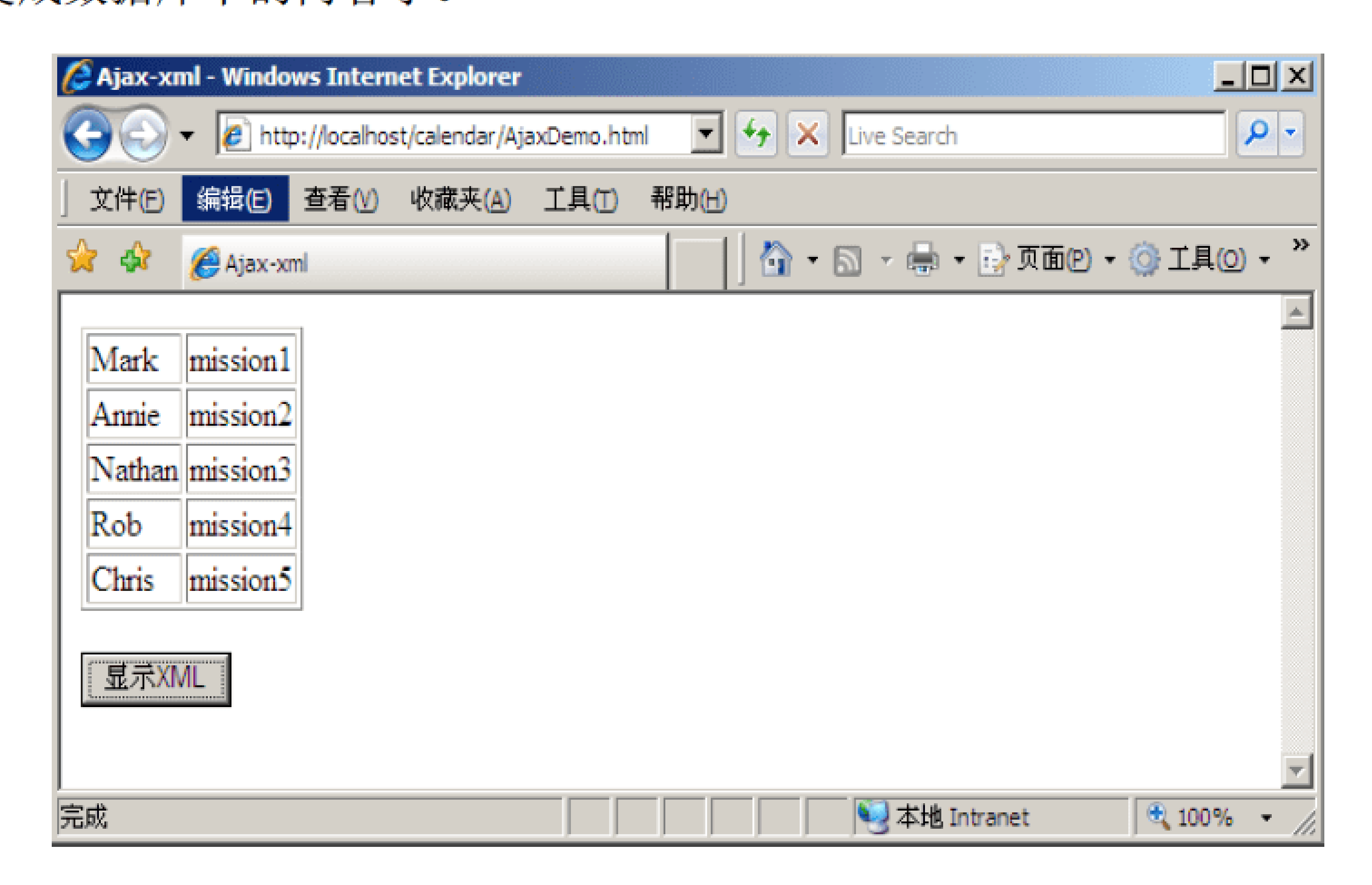


图 9.5　数据库驱动的 AJAX 应用

经过以上步骤，基本的 AJAX 应用流程就完整地实现了。网络上大量 AJAX 应用的基本结构和思路都是和本章介绍的相同，读者可以仔细体会。

9.3　日程表数据库设计

日程表的数据库设计的目的是为了存储提醒信息。它有两个功能，第一个是给管理者一个管理后台，便于管理及更新日程数据；第二个是前台访问者可以通过前端的页面获得最新的日程安排信息。了解清楚功能划分之后，下面来看具体的设计步骤。

（1）建立 calendar 数据库，SQL 语句如下：

```
CREATE DATABASE 'calendar' ;
```

（2）在 calendar 数据库中建立一个命名为 events 的消息事件表，结构如表 9.2 所示。创建表及相关字段的 SQL 语句如下：

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
```

```
-- 数据库: 'calendar'
--

-- --------------------------------------------------------

--
-- 表的结构 'events'
--

CREATE TABLE 'events' (
  'num' bigint(20) NOT NULL auto increment,
  'heading' varchar(45) NOT NULL default '',
  'date' date NOT NULL default '0000-00-00',
  'body' text NOT NULL,
  PRIMARY KEY  ('num')
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO INCREMENT=3
;
```

表 9.2　消息事件表

字　段	数 据 类 型	长　度	NULL	默　认　值	字 段 说 明
num	bigint	20	否	auto_increment	自增主键
heading	varchar	45			消息事件标题
date	date			Default '0000-00-00',	事件创建的日期
body	text				事件的说明

为了方便下面的程序调试，插入两条测试用的数据。

```
INSERT INTO 'events' VALUES (1, 'A great event!', '2010-06-14', 'The world
has never seen such a great event happening here today! This is just a sample
event, you can delete it now.');
INSERT INTO 'events' VALUES (2, 'A great event2', '2010-06-15', 'information
test');
```

注意：消息事件表的设计并不复杂，与以往通过 ID 序号查询不同的是，日程表模块所有的查询都是基于该表中的日期字段，这样设计的好处读者可以在下面的程序部分仔细体会。

9.4　日历程序设计

日程表的前端程序的核心是，通过 PHP 脚本生成一个当前日期的日历表格，再通过遍历数据中的消息事件，在有事件提醒的日期生成一个有“event”标识的链接，单击该链接就可以异步将对应的提醒信息显示在日历表格的下方。

9.4.1　PHP 生成的日历

PHP 生成日历是这个模块逻辑上最复杂的部分，下面来看如何生成一个标准的日历。日历展示的部分是将星期和日期通过一张二维表格输出到浏览器中的。日历表格的形式如图 9.6 所示。

Jun 2010						
Sun	Mon	Tue	Wed	Thur	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			
calendar						

图 9.6　日历表格样式

现在以 2010 年 6 月 14 日为例说明。6 月份共有 30 天，在这个日历表中共 5 行，首先需要确定当前所在的日期（当然也可以指定一个日期），比如 14 日，通过表格可以知道它在第 3 行对应星期一那列（第 3 周）。通常每个月第 1 天不是在表格开始的位置，位置并不固定，同样不固定的还有每个月的周数，有时候是 4 周，有时候是 5 周。这样，想要完成这个日历表格就需要确定一些基础的日期变量，下面具体来看需要设定的基础变量。

PHP 提供了 date()函数，该函数提供了丰富的日期处理功能。现在需要获得的数据有两个，第一个是当月的总天数；第二个是该月的第一天所在星期中的第几天，数字表示 0（表示星期天）到 6（表示星期六）。通过 date()函数可以很容易获得上面的数据。

```
<?php
//date()函数的基本用法
$year = date('Y');      //获得年份，例如 2006
$month = date('n');     //获得月份，例如 04
$day = date('j');       //获得日期，例如 3
?>
```

通过 mktime()函数和 date()函数获得当月的总天数。

```
//获得当月的总天数
$daysInMonth = date("t",mktime(0,0,0,$month,1,$year));
```

同样是组合使用 mktime()函数和 date()函数，获得该月的第一天所在星期中的第几天，数字表示 0（表示星期天）到 6（表示星期六）。

```
//获得每个月的第一天，例如 4
$firstDay = date("w", mktime(0,0,0,$month,1,$year));
```

现在重新来观察下这个日历表格，我们发现，表格中的值是从 1 到 x（当月的总天数）。我们可以形象地通过一个坐标来表示每一个位置(x,y)，表格的起始位置是(0,0)，结束位置是(5,4)，当月第一天的起始位置是(0,3)。现在来创建一个二维数组来存储对应的日期。

（1）需要获得表格的数目。

```
//计算数组中的日历表格数
$tempDays = $firstDay + $daysInMonth;
```

（2）需要算出该月一共有几周（即表格的行数）。

```
//获得表格行数
$weeksInMonth = ceil($tempDays/7);
```

（3）在获得行数的变量之后，创建一个二维数组用来存放日期信息，代码如下：

```
<?php
for($j=0;$j&lt;$weeksInMonth;$j++)
{
    for($i=0;$i&lt;7
    ;$i++)
    {
        $counter ++;
        $week [$j] [$i] = $counter;
    }
}
?>
```

（4）细心的读者可能发现，上面的代码其实是一个雏型，因为它处理出来的数据并没有过滤那些空白的表格单元。下面来继续改造下这段代码，加入过滤和构造的部分。改造后的代码如下：

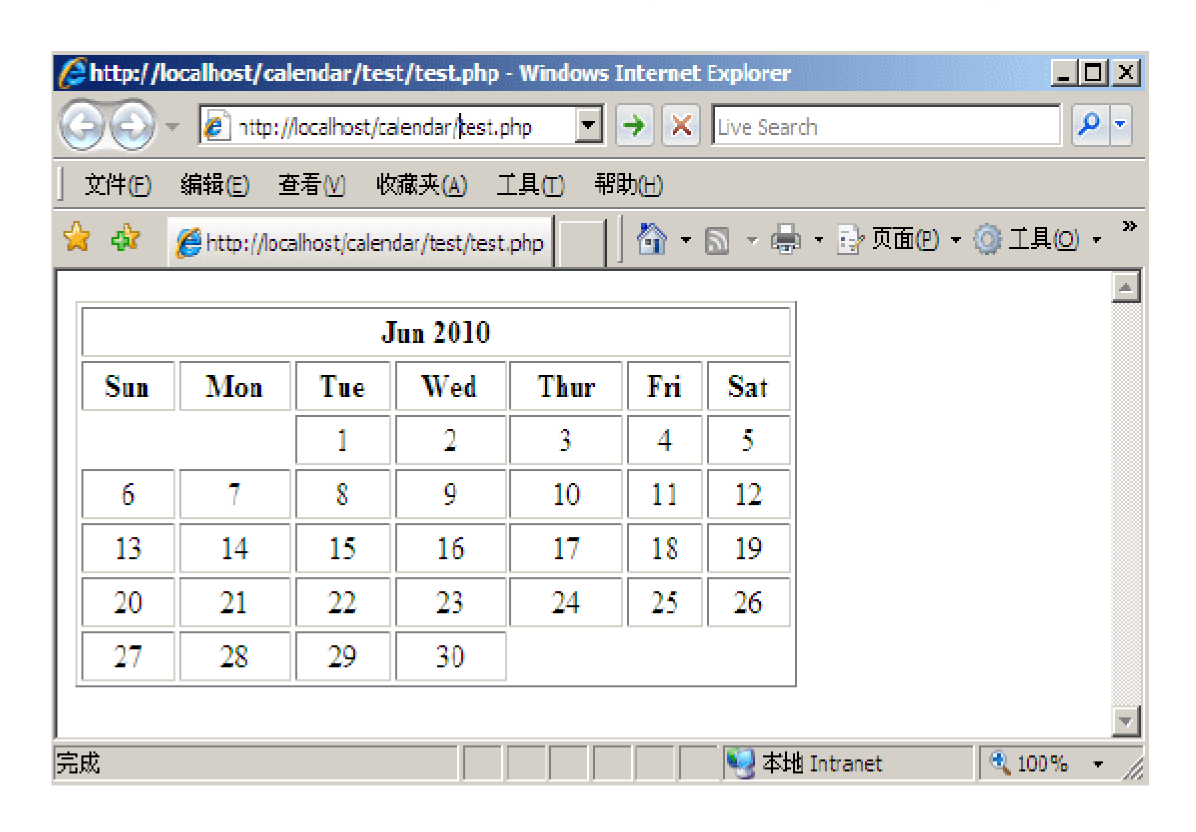
```
<?php
//创建日期二维数组
for($j = 0; $j < $weeksInMonth; $j ++) {
    for($i = 0; $i < 7; $i ++) {
        $counter ++;
        $week [$j] [$i] = $counter;
        //日期偏移量
        $week [$j] [$i] -= $firstDay;
        if (($week [$j] [$i] < 1) || ($week [$j] [$i] > $daysInMonth)) {
            $week [$j] [$i] = "";
        }
    }
}
?>
```

（5）在获得正确的二维数组之后，就可以通过 foreach()函数将存储的日期信息遍历出来，同时插入 HTML 标签创建日期，代码如下：

```
<?php
$year = date ( 'Y' );   //获得年份，例如： 2006
$month = date ( 'n' ); //获得月份，例如： 04
$day = date ( 'j' );   //获得日期，例如： 3
$firstDay = date ( "w", mktime ( 0, 0, 0, $month, 1, $year ) );
                                       //获得当月第一天
$daysInMonth = date ( "t", mktime ( 0, 0, 0, $month, 1, $year ) );
                                       //获得当月的总天数
//echo $daysInMonth;
$tempDays = $firstDay + $daysInMonth;  //计算数组中的日历表格数
$weeksInMonth = ceil ( $tempDays/7 );  //算出该月一共有几周（即表格的行数）
//创建一个二维数组
for($j = 0; $j < $weeksInMonth; $j ++) {
    for($i = 0; $i < 7; $i ++) {
        $counter ++;
        $week [$j] [$i] = $counter;
        //offset the days
        $week [$j] [$i] -= $firstDay;
        if (($week [$j] [$i] < 1) || ($week [$j] [$i] > $daysInMonth)) {
            $week [$j] [$i] = "";
        }
    }
}
```

```
?>
<script type="text/JavaScript" src="calendar.js"></script>
<table width="400" border="1" cellpadding="2" cellspacing="2">
    <tr>
        <th colspan='7'>
            <?php
            echo date ( 'M', mktime ( 0, 0, 0, $month, 1, $year ) ) . ' ' .
            $year;
            ?>
        </th>
    </tr>
    <tr>
        <th>Sun</th>
        <th>Mon</th>
        <th>Tue</th>
        <th>Wed</th>
        <th>Thur</th>
        <th>Fri</th>
        <th>Sat</th>
    </tr>
<?php
foreach ( $week as $key => $val ) {
    echo "<tr>";
    for($i = 0; $i < 7; $i ++) {
        echo "<td align='center'>" . $val [$i] . "</td>";
    }
    echo "</tr>";
}
?>
```

（6）在浏览器中运行上面的代码，得到一个完整的日历表格，效果如图 9.7 所示。



Jun 2010						
Sun	Mon	Tue	Wed	Thur	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

图 9.7　PHP 生成日历

9.4.2　日程表 XML 文档

在 9.1 节中介绍过 AJAX 的运行模式，其中的数据传输是通过解析标准的 XML 格式文档进行的，在日程表模块沿用这种模式。首先将数据中的信息遍历出来，然后根据日期获得对应的内容。XML 格式的标准已经在 9.2 节中介绍过了，日程表模块的 XML 文档很简单，样式如下：

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
```

```
<content>
<![CDATA[ ]]>
</content>
</response>
```

其中，Content 节点部分的内容用来存放数据，创建一个文件名为 xml.php 的脚本，来读取数据中的信息并输出成 XML 格式。该文件的内容如下：

```
<?php
header ( 'Content-Type: text/xml' );                    //header 函数声明文件类型
echo '<?xml version="1.0" encoding="utf-8" ?>';    //输出 XML 格式标签头
$date = $_GET ['event'];
$db = mysql_connect ('localhost', 'root', '198251');  //数据库连接信息
mysql_select_db ( 'calendar' );                         //选择指定的数据库
$sql = "SELECT 'body' FROM 'events' WHERE 'date' = '" . $date . "' LIMIT
0 , 1 ";
$result = mysql_query ( $sql );
$rs = mysql_fetch_row ( $result );
//输出<response>标签
echo ' <response>
<content>
<![CDATA[
    <tr>
        <th colspan="7" >' . $rs [0] . '</th>
    </tr>
    ]]>
</content>
</response>';
?>
```

在浏览器中运行如下链接，效果如图 9.8 所示。

```
http://localhost/calendar/xml.php?month=&year=&event=2010-06-15
```

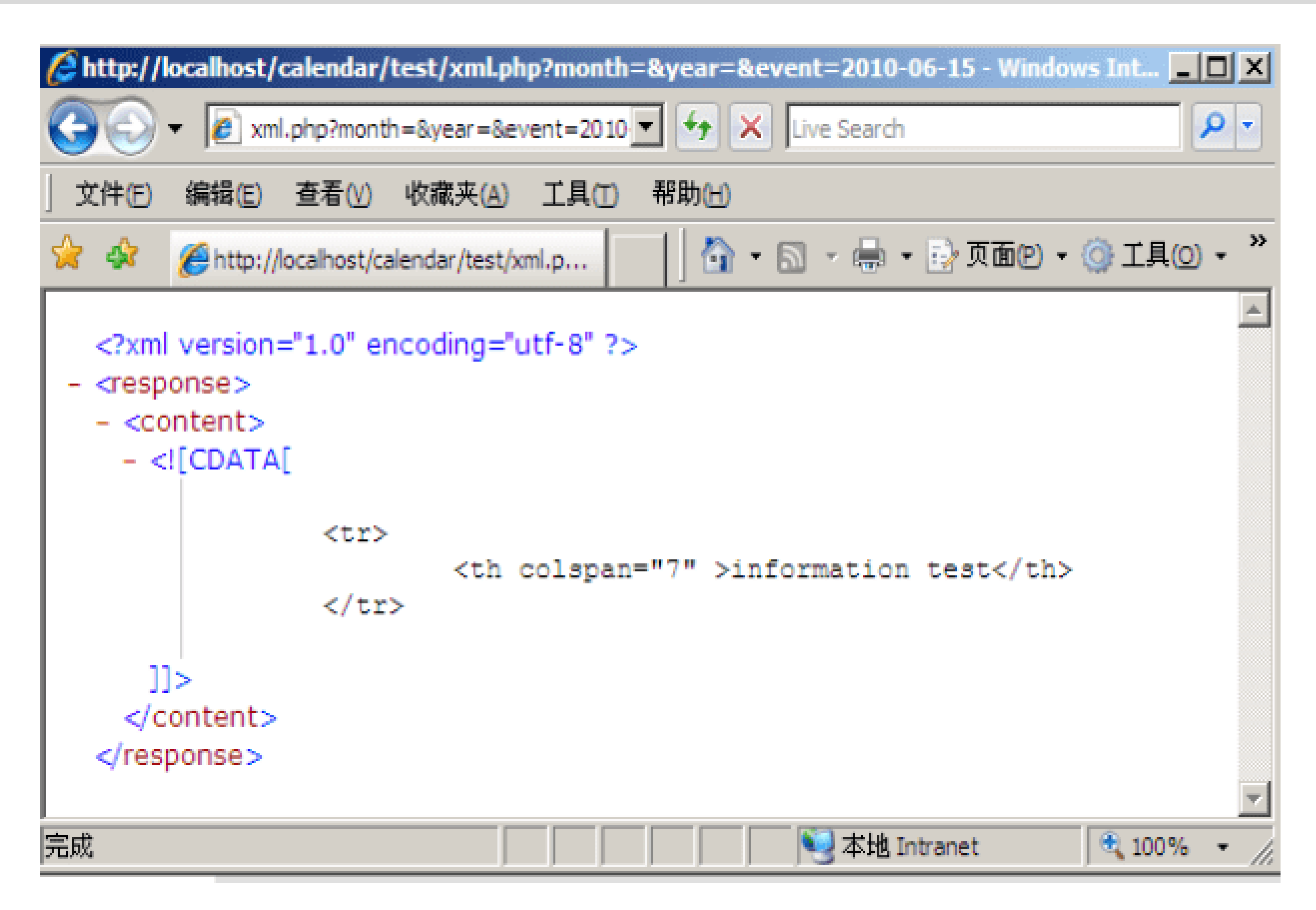


图 9.8　PHP 生成日程表 XML 文档

9.4.3　日历的完美实现

通过前两节的准备后，本节将独立的部分整合并完成日程表程序。日程表程序由 AJAX 配置文件（calendar.js）、PHP 日历文件（calendar.php）、日程表 XML 文档（xml.php）3 个部分组成。

1．AJAX配置文件

创建文件名为 calendar.js 的文件，文件内容如下：

```
var req;
//日历函数
function navigate(month,year,evt) {
    var url = "xml.php?month="+month+"&year="+year+"&event="+evt;
    //alert (url);
    if(window.XMLHttpRequest) {
        req = new XMLHttpRequest();
    } else if(window.ActiveXObject) {
        req = new ActiveXObject("Microsoft.XMLHTTP");
    }
    req.open("GET", url, true);
    req.onreadystatechange = callback;
    req.send(null);
}
//异步数据通知
function callback() {
    if(req.readyState == 4) {
        var response = req.responseXML;
        var resp = response.getElementsByTagName("response");
        document.getElementById("calendar").innerHTML = resp[0].get-
        ElementsByTagName("content")[0].childNodes[0].nodeValue;
        //getObject("calendar").innerHTML = resp[0].getElementsByTagName
        ("content")[0].childNodes[0].nodeValue;
        //alert('hi');
    }
}
//获得元素节点
function getObject(obj) {
    var o;
    if(document.getElementById) o = document.getElementById(obj);
    else if(document.all) o = document.all.obj;
    return o;
}
```

【代码解读】

其中，navigate()为入口函数，用来初始化 AJAX 应用，然后调用 callback()函数将从 xml.php 获得的数据插入 HTML 代码中。

2．PHP日历文件

打开 calendar.php 文件之前，该文件完成了处理当前日期信息并生成日程表页面。下面对该文件进行改造。

（1）添加从数据库中获得信息内容，代码如下：

```
<?php
$db = @mysql_connect($localhost,$user,$pwd) ;  //数据库连接信息
mysql_select_db('calendar');                   //选择数据库
$sql = "SELECT num,date FROM 'events' LIMIT 0 , 30 ";
$result = mysql_query($sql);
while ($rw = mysql_fetch_row($result)) {
        $links[]=$rw[1];                       //将时间串取出存入数组
}
?>
```

【代码解读】

以上的代码将数据库中的事件信息遍历出来，并处理成一个数组，便于下面生成日程表时做比对，同时请注意替换数据库链接信息。

（2）现在改造生成日程表部分的代码，同时在页面代码中引入 AJAX 的配置文件，用来处理提交的数据。

```
<script type="text/JavaScript" src="calendar.js"></script>
<table width="400" border="1" cellpadding="2" cellspacing="2">
    <tr>
        <th colspan='7'>
            <?php echo date('M', mktime(0,0,0,$month,1,$year)).' '.$year;?>
        </th>
    </tr>
    <tr>
        <th>Sun</th>
        <th>Mon</th>
        <th>Tue</th>
        <th>Wed</th>
        <th>Thur</th>
        <th>Fri</th>
        <th>Sat</th>
        </tr>
<?php
foreach ($week as $key => $val) {
    echo "<tr>";
    for ($i=0;$i< 7;$i++) {
    //日期匹配
    if (in array("2010-06-".$val[$i],$links)) {
echo "<td align='center'>
<a  href='JavaScript:navigate(\"\",\"\",\"2010-06-".$val[$i]."\")'>Event
</a>".$val[$i]."</td>";
        }else{
            echo "<td align='center'>".$val[$i]."</td>";
        }
    }
        echo "</tr>";}
?>
```

注意： 当一个日期有提醒事件时，就需要在日程表格中加入一个提示标识并添加链接，通过 in_arrary()就可以判断出来该日期是否在数组中，再通过 if 判断就可以加入事件标识了。

（3）在文件的底部加入一个空的表格用来获得数据库信息（生成 HTML 标签），代码如下：

```
<tr id="calback">
<th id="calendar" colspan='7'>
    calendar
</th>
</tr>
</table>
```

（4）至此，calendar.php 的改造就完成了，在浏览器中运行该文件效果如图 9.9 所示。其中可以看到，在数据库设置事件的日期表格有“event”标识，并可以通过链接驱动 AJAX 文件。

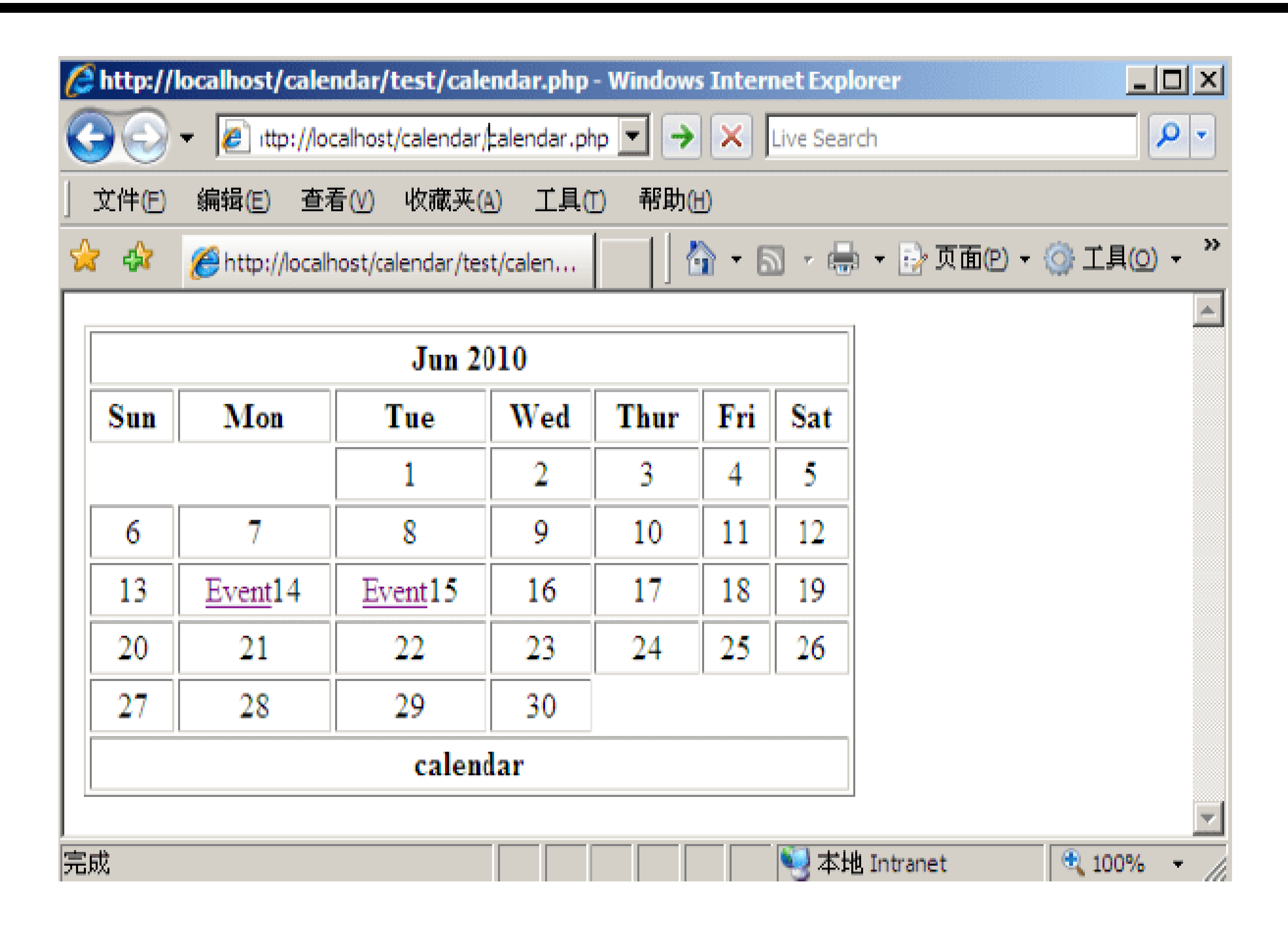


图 9.9　PHP 日程表

3．日程表事件提醒

现在回顾并整理日程表事件提醒程序的运行流程：

（1）首先通过 PHP 日程表文件（calendar.php）生成当月的日历表格；

（2）再比对数据库中的事件日期，将有事件的日期添加链接。链接形式如下：

```
<a href='JavaScript:navigate("","","2010-06-14")'>Event</a>
```

【代码解读】

当点击该链接时，就会触发 AJAX 的配置文件中的 navigate()函数。该文件会读取 xml.php 文件中对应日期的 XML 格式信息，该信息由 xml.php 文件从数据库中读取信息后生成。当数据被更新后会通过 XMLHttpRequest 状态位的改变，异步的将数据加入到 PHP 日程表页面指定的位置完成数据更新。

现在通过浏览器打开 calendar.php 文件，选择一个有事件提醒的日期并点击链接，效果如图 9.10 所示。

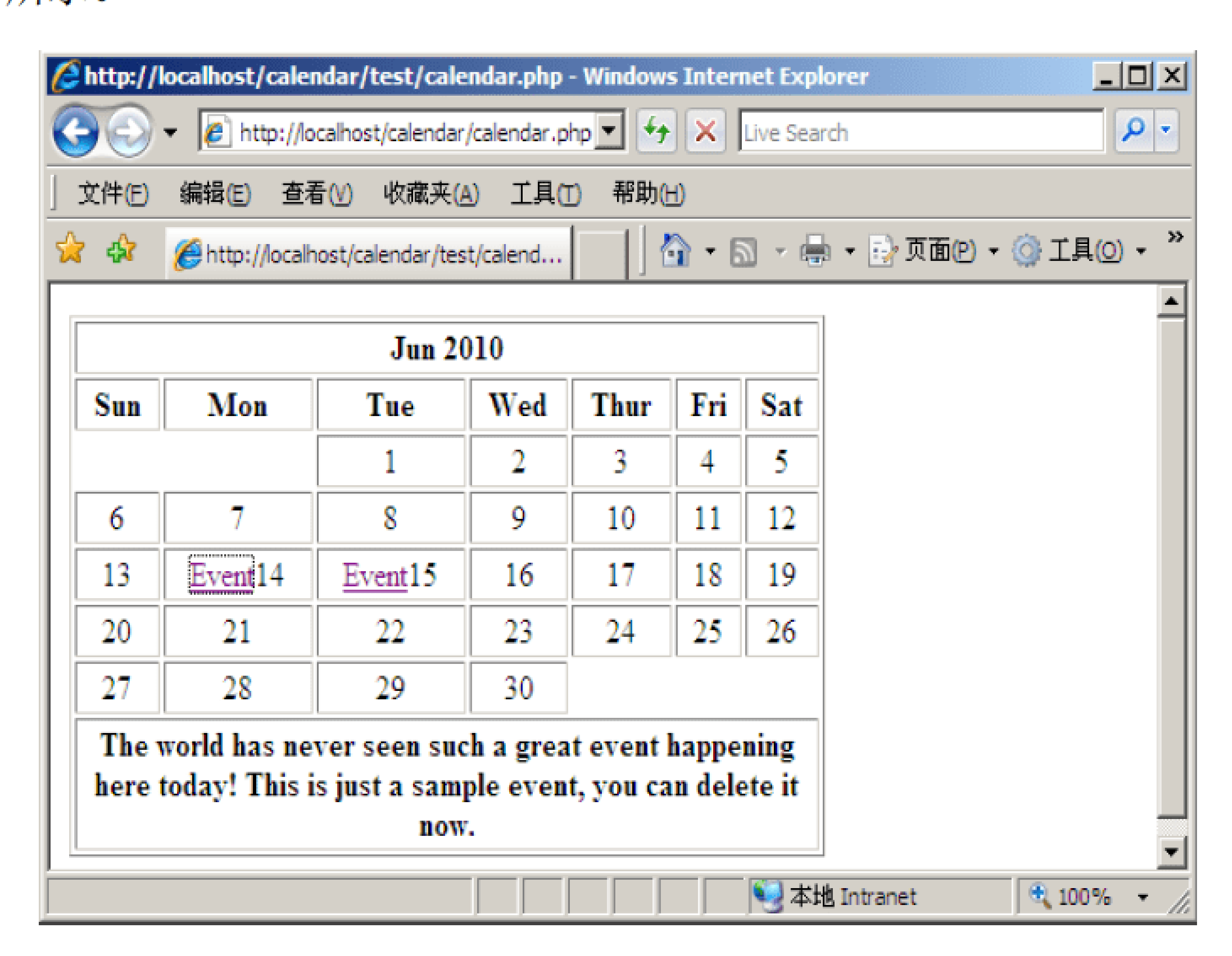


图 9.10　PHP 日程表数据更新

通过图 9.10 可以看到，从数据库中获得的数据被更新到页面上了。同时，由于是异步更新，只是更换了数据的部分，其他页面元素没有更新，最大程度的减少了代码的更新量，从而获得非常高的响应速度，用户体验也会极大地提高。

9.5　日程表管理程序

前面已经完成了日程表的前台程序部分，但是前端所看的事件消息是由管理员通过后台发布并负责维护的。为了便于管理员发布及管理事件信息需要创建日程表的管理后台，该部分程序按照功能分为添加事件部分和管理（编辑、删除）事件部分。以上功能是日程表程序的基础管理功能。

9.5.1　添加事件界面

日程表提醒事件信息是由后台管理员发布的，现在创建一个 calendar_admin.php 文件，它将提供给管理员添加事件的界面，同时也是后台管理程序的入口文件。按照管理功能区分这个界面分为操作面板部分和消息表单。

操作面板部分包括添加事件、编辑/删除事件。消息表单包括日期框（Date/Time）、事件标题（Heading）、事件内容（Event）。下面来看具体的代码：

```
<title>AJAX PHP Calendar Admin Tool</title>
<link rel="stylesheet" type="text/css" href="calendar admin.css" />
<script src="calendar1.js" type="text/JavaScript"></script>
<script type="text/JavaScript">
   var c = new Calendar("c");
   document.write(c);
</script>
</head>
<body>
<div id="main">
    <div id="top">
        <h3>AJAX Calendar Admin</h3>
        <div class="menu">
            <a href="calendar admin.php">New Event</a> | <a href="list.php">
            Edit / Delete Event</a>
        </div>
    </div>
    <div id="bottom">
<h3>New Event</h3>
         <form id="form1" runat="server">
        <div>
            <div class="field">
            <span class="label">Date / Time:</span>
            <span class="box">
                <input type="text" name="date"  ID="TextBox1" onfocus=
                "calendar()"/>
               <p style="margin: 4px 0px 0px 0px">   </p>
            </span></div>
            <div class="field">
                <span class="label">Heading:</span>
                    <span class="box">
```

```
                        <input type="text" size="50" name="heading" />
                    </span>
            </div>
            <div class="field">
                <span class="label">Event:<br />
                    <span style="font-size:9px">(HTML is OK)</span>
                    </span><span class="box">
                <textarea name="event" cols="50" rows="6"></textarea>
                    </span>
            </div>
        </div>
        <br />
        <div class="field">
            <span class="label"> </span>
                <span class="box">
                    <input type="submit" value="Add Event" />
                </span>
        </div>
        </form>
</div>
</div>
<div id="calwin">
    <div class="bar">
        <a href="JavaScript:hideCal()">
            <img src="/calendar/images/calClose.gif" alt="close" />
        </a>
    </div>
    <div id="calback">
        <div id="calendar"></div>
    </div>
</div>
</body>
</html>
```

保存该文件，并通过浏览器预览，效果如图 9.11 所示。

说明：calendar1.js 文件是一个 JavaScript 的日历程序，在页面表单中通过 onfocus="calendar()"事件触发用来添加事件的日期。

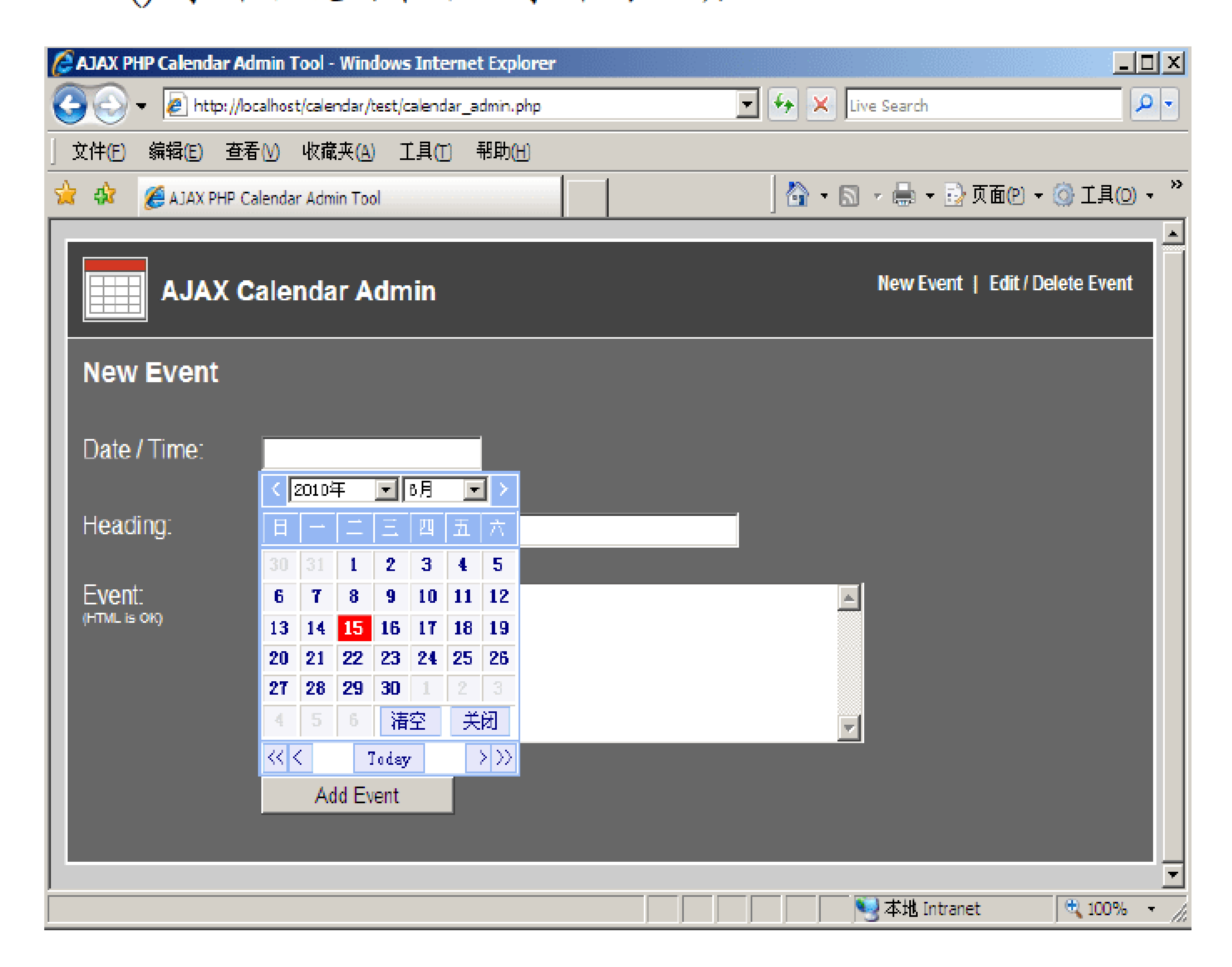


图 9.11　日程表事件添加界面

9.5.2　添加事件程序

添加事件程序是管理员最常用的一个功能，该程序需要从表单中获得字段信息，然后写入数据。核心部分是获得正确的表单信息及写入数据库的过程。通过前几章的学习读者对这类程序已经非常熟悉了，下面来看具体的代码实现。

```
<?php
    //当表单传递过来的变量全部不为空才继续执行
    if(!empty($_POST['date']) and !empty($_POST['heading']) and !empty($_POST['event'])){
        $db = mysql_connect('localhost', 'root', '198251'); //数据连接信息
        mysql_select_db('calendar');                        //选择数据库
        $sql="INSERT INTO 'calendar'.'events' ('num' ,'heading' ,'date' ,'body' )
        VALUES (NULL , '".$_POST['heading']."', '".$_POST['date']."', '".$_POST['event']."');";
        $result = mysql_query($sql);
        }else{
        echo '<script type="text/JavaScript">alert(\'请填写完整信息\')</script>';
        exit;
    }
?>
```

【代码解读】

为了避免误操作将无效的数据写入数据库，对表单提交过来的信息做有效性的判断，这样可以避免数据冗余和产生无效数据。

将以上代码加入到 calendar_admin.php 文件的最下方，在浏览器中执行测试当输入信息不完整时的异常处理，处理结果如图 9.12 所示。

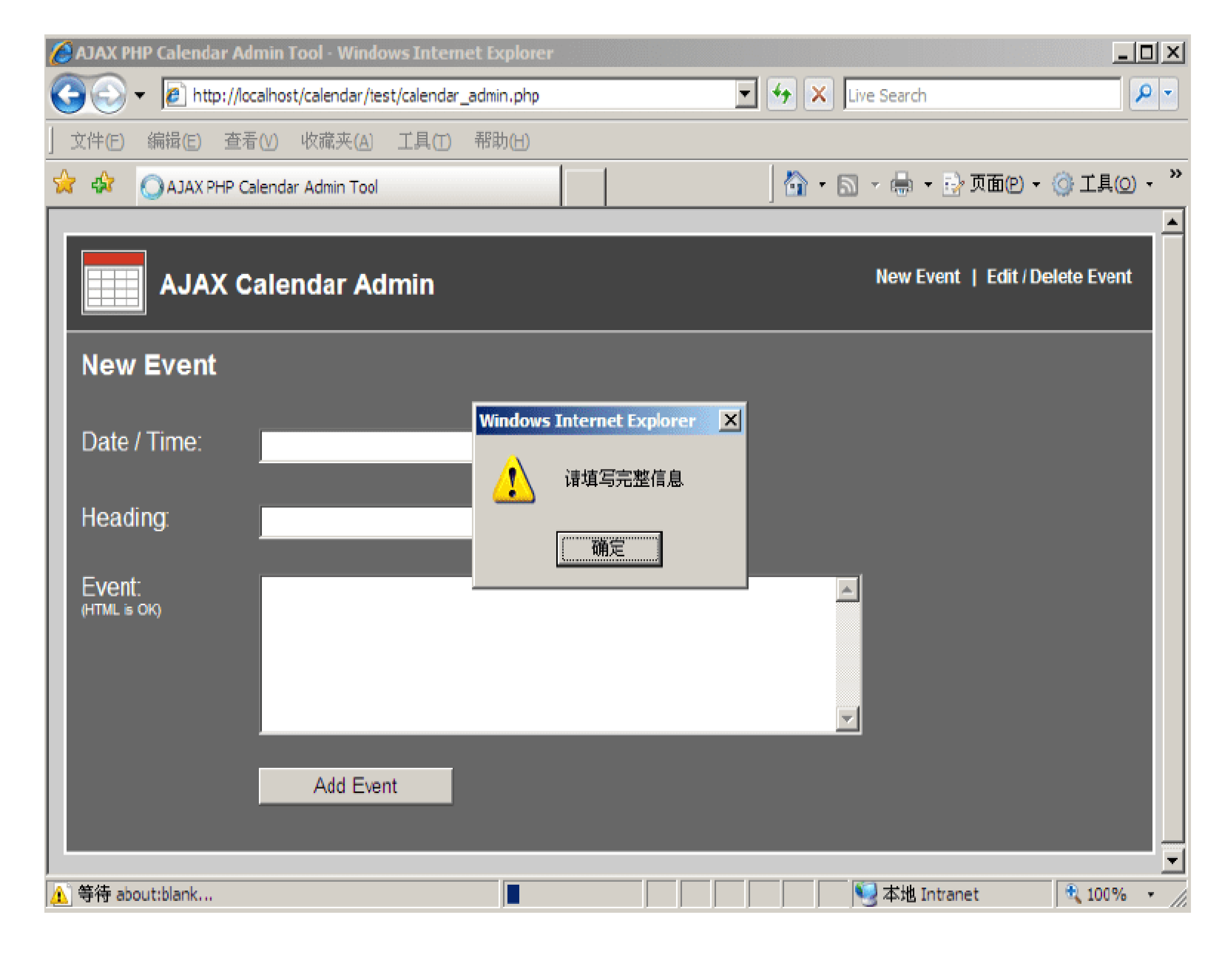


图 9.12　输入信息不完整的异常处理

9.5.3　管理事件界面

当管理员想要更新某个事件消息，或者对过期的消息进行清理，就需要在事件管理界面进行操作。此功能共分为两个页面，即事件汇总页面（list.php）和编辑/删除页面（op.php）。

下面介绍事件汇总页面（list.php）的设计代码，该文件代码如下：

```
<title>AJAX PHP Calendar Admin Tool</title>
</head>
<body>
<div id="main">
    <div id="top">
        <h3>AJAX Calendar Admin</h3>
        <div class="menu">
            <a href="?f=new&amp;sf=list">New Event</a> | <a href="?f=edit&
            amp;sf=list">Edit / Delete Event</a>
        </div>
    </div>
    <div id="bottom">
<h3>Edit / Delete Event</h3>
<form name="f" method="post" action="op.php">
        <div id="fields">
            <div class="field"><span class="label">Event:</span><span
            class="box">
            <select name="num">
                <option value=""></option>
                <option value="2">[ 06-15-2010 ] A great event2</option>
                <option value="1">[ 06-14-2010 ] A great event!</option>
                <option value="9">[ 06-09-2010 ] sfdsf</option>
            </select>
            <input type="submit" value="    Edit Event    " /></span></div>
        </div>
</form>
```

编辑/删除页面（op.php）可以在 calendar_admin.php 页面的基础上稍做处理即可，页面元素并无变化。

9.5.4　管理事件程序

打开 list.php 事件汇总页面，再插入 PHP 脚本程序，将数据库中的事件信息全部遍历出来，然后生成<option>标签显示出来。具体的代码实现如下：

```
<title>AJAX PHP Calendar Admin Tool</title>
<!--<link rel="stylesheet" type="text/css" href="calendar_admin.css" />
-->
</head>
<body>
<div id="main">
    <div id="top">
        <h3>AJAX Calendar Admin</h3>
        <div class="menu">
            <a href="?f=new&amp;sf=list">New Event</a> | <a href="?f=
            edit&amp;sf=list">Edit / Delete Event</a>
        </div>
    </div>
```

```
    <div id="bottom">
<h3>Edit / Delete Event</h3>
<form name="f" method="post" action="op.php">
        <div id="fields">
            <div class="field"><span class="label">Event:</span><span
            class="box">
            <select name="num">
                <option value=""></option>
                <?php
    $db = mysql_connect('localhost', 'root', 'pwd');   //数据库连接信息
    mysql_select_db('calendar');                       //选择数据库
    $sql="SELECT *FROM 'events' LIMIT 0 , 30";
    $result = mysql_query($sql);
    //$info=mysql fetch array($result);                //过程调试语句
    $num=mysql_num_rows($result);                      //获得结果集行数
    for ($i=0;$i<$num;$i++){
        $info=mysql fetch array($result);
    echo  "<option  value='".$info['num']."'>[  06-15-2010  ]  '".$info
    ['heading']."'</option>";
        }
    ?>
            </select>
            <input type="submit" value="   Edit Event    " /></span></div>
        </div>
</form>
</div>
</div>
</body>
</html>
```

保存文件，在浏览器中运行该程序如图 9.13 所示。可以看到，数据库中消息事件标题已经在页面中显示出来了。

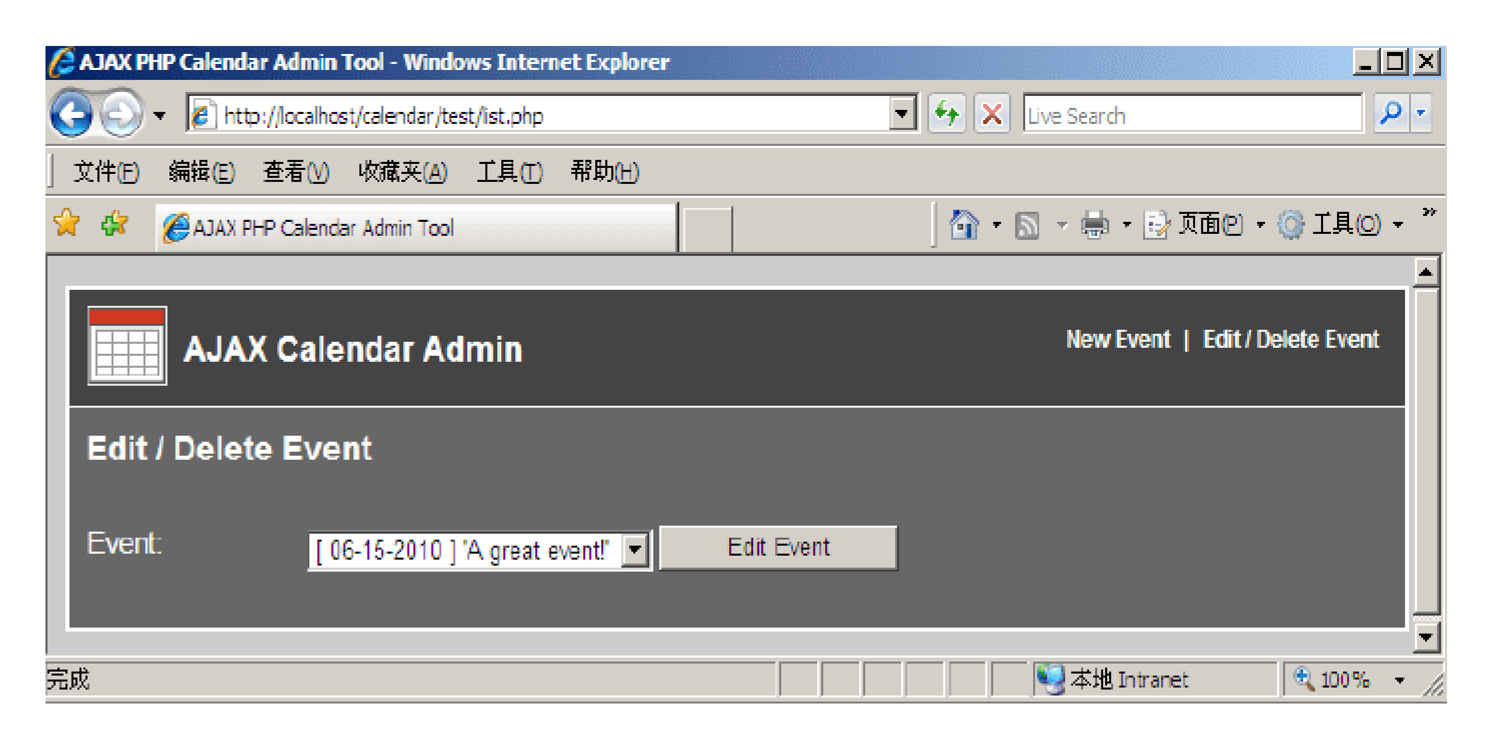


图 9.13　事件汇总程序

当单击“Edit Event”按钮后，将数据以 POST 的方式提交到编辑/删除程序（op.php）做处理。编辑信息的程序流程如下：

（1）根据 num 标签的值将该条数据从数据库中取出，并输出到页面中；

（2）当文本框中的内容做了改动，再通过 update 操作更新数据库中 num 对应的内容，删除功能与编辑操作类似。

详细的代码实现如下所示。

```
<?php
    if(!empty($_POST['num'])){                           //判断 num 值是否为空
        $db = mysql_connect('localhost', 'root', 'pwd');   //数据库连接信息
```

```
        mysql_select_db('calendar');                        //选择数据库
        $sql="SELECT * FROM 'events' WHERE 'num'='".$_POST['num']."'";
        $result = mysql_query($sql);
        $info=mysql_fetch_array($result);                    //获得结果集
        }else{
        echo '<script type="text/JavaScript">alert(\'该事件不存在\')</script>';
        exit;
    }
?>
<title>AJAX PHP Calendar Admin Tool</title>
<!--<link rel="stylesheet" type="text/css" href="calendar_admin.css" />
-->
<script src="calendar1.js" type="text/JavaScript"></script>
<script type="text/JavaScript">
   var c = new Calendar("c");
   document.write(c);
</script>
</head>
<body>
<div id="main">
    <div id="top">
        <h3>AJAX Calendar Admin</h3>
        <div class="menu">
            <a href="calendar_admin.php">New Event</a> | <a href="list.
            php">Edit / Delete Event</a>
        </div>
    </div>
    <div id="bottom">
<h3>Edit / Delete Event</h3>
         <form id="form1" runat="server" action="calendar_admin.php" method=
         "post">
        <div>
        <img src="/calendar/images/calDelete.gif" /> <a href="del.php?num
        =<?php echo $info['num']; ?>" onClick="return confirm('确认删除该消
        息吗？')" style="color:#F26343;font-weight:bold">Delete Event</a>
            <div class="field">
            <span class="label">Date / Time:</span>
            <span class="box">
                <input type="text" name="date" value="<?php echo $info
                ['date']; ?>" />
               <p style="margin: 4px 0px 0px 0px">  </p>
            </span></div>
            <div  class="field"><span  class="label">Heading:</span><span
                class="box"><input  type="text"  size="50"  name="heading"
                value="<?php echo $info['heading']; ?>"/></span></div>
            <div class="field"><span class="label">Event:<br /><span
            style="font-size:9px">(HTML is OK)</span></span><span class=
            "box"><textarea name="event" cols="50" rows="6">"<?php echo
            $info['body']; ?>"</textarea></span></div>
        </div>
        <br />
        <div class="field"><span class="label"> </span><span class=
        "box"><input type="submit" value="   Save Event   " /></span></div>
        </form>

    </div>
</div>
<div id="calwin">
    <div class="bar"><a href="JavaScript:hideCal()"><img src="/calendar/
    images/calClose.gif" alt="close" /></a></div>
    <div id="calback">
```

```
        <div id="calendar"></div>
    </div>
</div>
</body>
</html>
<?php
    //当表单传递过来的变量全部不为空才继续执行
    if(!empty($_POST['date']) and !empty($_POST['heading']) and !empty($_POST['event'])){
        $db = mysql_connect('localhost', 'root', '198251');
        mysql_select_db('calendar');
        $sql="UPDATE 'calendar'.'events' SET 'heading' = '".$_POST['heading']."','date' = '".$_POST['date']."','body' = '".$_POST['event']."' WHERE 'events'.'num' ='".$_POST['num']."' LIMIT 1 ;";
        $result = mysql_query($sql);
        if($result){
        echo '<script type="text/JavaScript">alert(\'更新成功\')</script>';
        }
}
?>
```

有了前面的基础删除功能并不难实现，观察删除链接的特征：

```
http://localhost/calendar/del.php?num=9
```

很显然，只需要在数据中删除 num 字段值为 9 的行就可以了，创建 del.php 文件用来从数据库中删除过期数据。del.php 代码如下：

```
<?php
    if(!empty($_GET['num'])){
        $db = mysql_connect('localhost', 'root', 'pwd');   //数据库连接信息
        mysql_select_db('calendar');                       //选择数据库
        //删除操作 SQL 语句
    $sql="Delete From 'calendar'.'events' WHERE 'events'.'num' ='".$_GET['num']."' LIMIT 1 ;";
        $result = mysql_query($sql);
        if($result){
        echo '<script type="text/JavaScript">alert(\'successful: ) \')</script>';
        }
}
?>
```

9.6　小　　结

AJAX 模式的应用程序因为有诸多的优点，所以在近年来被广泛地应用，尤其在用户体验方面的优势更是明显。本章首先引入 AJAX 的基本概念。第 9.2 节是对前一节知识的具体应用，介绍了 AJAX 中服务端脚本的内容，这部分又可以细分为 AJAX 与 XML 文件进行交互式通信、HP 生成 XML 和数据库驱动的 AJAX 3 个部分。第 9.4 节和 9.5 节介绍了 PHP 生成日历程序和日程表管理程序，这两节具体地介绍了日程表程序的设计和实现方法。

本章的重点内容是 AJAX 的实现原理和其中涉及的各个组成部分。AJAX 模式对很多读者可能有些陌生，通过本章的学习读者可以构建一个简单 AJAX 应用程序，其他更复杂的 AJAX 应用程序都是以此为基础的。

第 10 章　聊天室模块（Frameset 页面框架+Session）

在这个信息极其发达的时代，人与人之间交流的形式也日渐多样化。目前可以用来进行信息交流的形式也非常多，但是其中有些交流方式因为不够便捷，或者有局限性和费用的问题，都不能很好地解决实时沟通交流的问题。网络聊天室这种形式的出现很好地解决了这个问题。“交谈”的双方不需要聚集在同一地点，各自在不同的地点就可以在聊天室中发言互相讨论。同时聊天室具有信息丰富和互动性强的特点，因此这种形式在网络上被广泛地使用。

本章着重介绍聊天室的业务流程及技术实现原理。读者通过本章的学习，可以了解到聊天室程序的逻辑结构设计和程序组成，在具体的技术实现细节方面，读者需要了解 PHP 中的服务器变量（Session）的实际运用、交互式程序的实现方法等。

本章主要涉及的知识点如下。

- 聊天室发言的两种模式：表情加文字输入模式和选择动作描述模式。
- PHP 中的 Session 变量：属于 PHP 中的预定义变量在服务器运行。
- mysql_query(set names utf-8)函数：设置接受数据的字符集。
- META 实现自动刷新页面：通过设置 content、url 控制跳转的秒数和跳转的地址。
- session_destroy()函数：通过这个函数可以销毁整个 Session 文件。

10.1　聊天室基础设计

聊天室的业务逻辑过程是将发言者所要表达的信息发送到指定的显示区域，同时提供在线的网友浏览。编辑发言信息是最主要的一个环节，可以使用文字输入的方式也可以选择一些固定的动作描述或图片表达，发言时带一些表情会更加生动有趣。常见的聊天室有两种发言模式，一种是表情加文字输入，另一种是选择动作描述。下面将详细介绍这两种模式，同时还将介绍聊天室的主要功能。

10.1.1　主要实现功能描述

聊天室具有很强的互动功能，这个特点可以很好地满足网友互动交流的需要。一个标准的聊天室模块程序需要实现的主要功能如下。

- 选择聊天的主题：根据不同的需要选择不同的聊天主题。
- 编辑聊天的内容：编辑待发言的信息。
- 显示聊天信息：可以查看聊天室内的全部发言。

❑ 聊天信息的存储：将聊天信息按照时间顺序写入数据库中。

10.1.2　业务流程描述

聊天室业务流程是指从一个访客进入聊天室到聊天结束的整个过程，它也是确定各个程序文件需要完成功能的模型基础，业务流程如下：

（1）用户向服务器发出访问请求。

（2）服务器根据向用户传送欢迎页面，并提供选择的聊天主题。

（3）进入聊天室页面。

（4）编辑聊天信息，发送到数据库中存储。

（5）从数据库提取发言信息显示在固定区域。

（6）结束聊天退出聊天室。

通过图 10.1 可以更直观地了解整个过程中程序的工作流程。

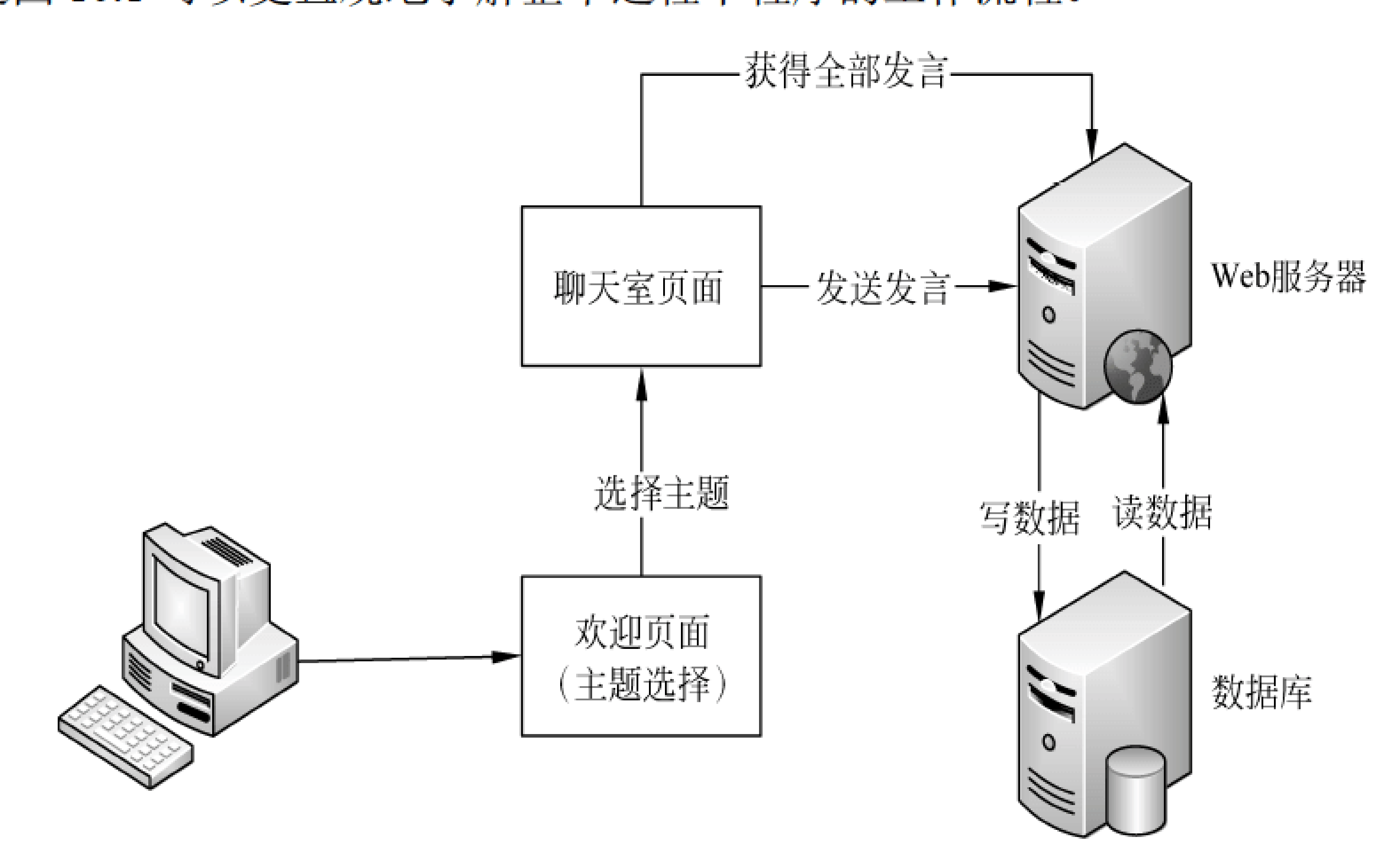


图 10.1　程序工作流程图

10.1.3　逻辑结构设计

根据对系统架构的描述，各个文件之间的逻辑关系已经非常清晰，现在通过一个逻辑框图梳理下逻辑，聊天室模块的系统逻辑结构，如图 10.2 所示。

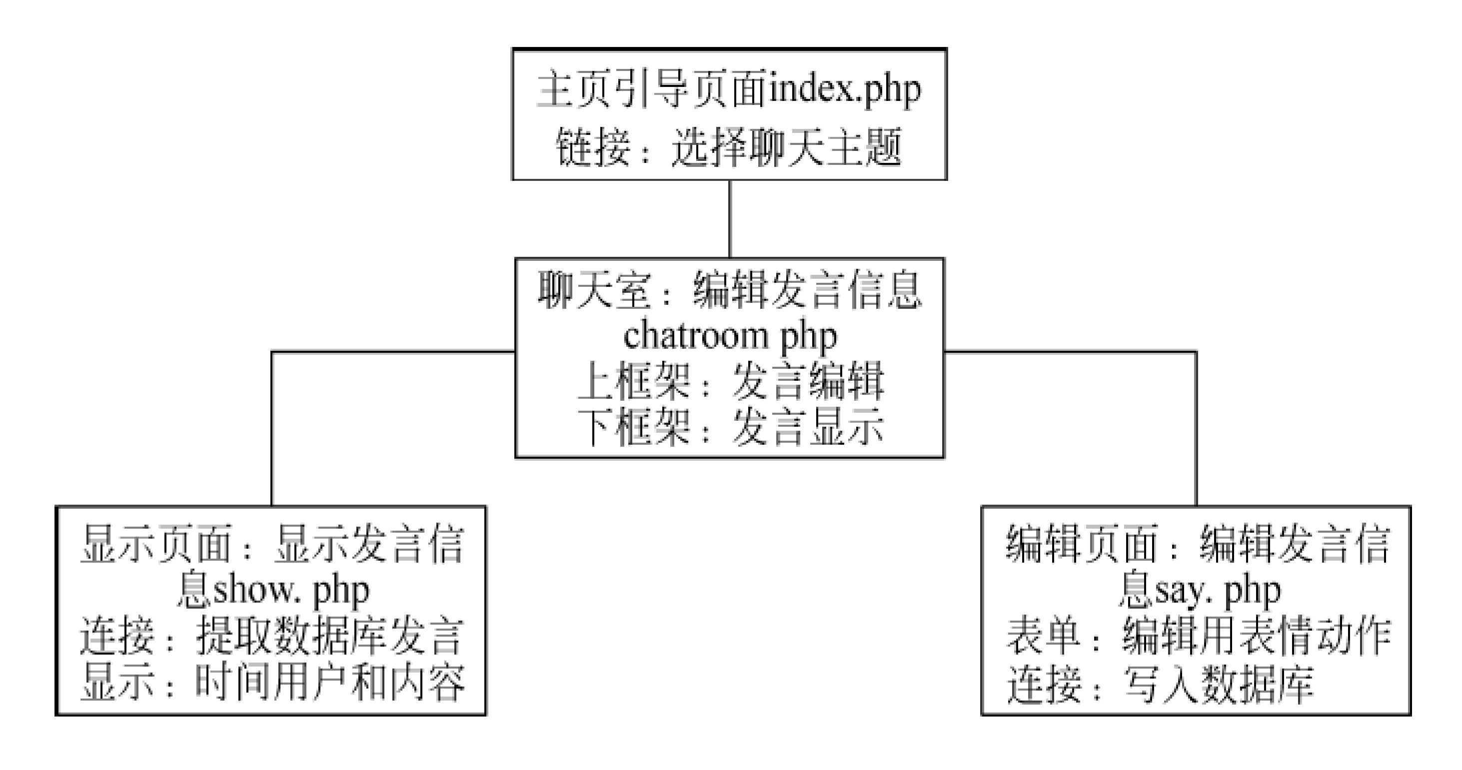


图 10.2　聊天室系统逻辑框图

10.2　聊天室数据库设计

聊天室部分的数据库设计中只有一个聊天信息表（text），用来记录聊天室中访问者的发言、表情及动作信息。由于整个聊天的过程都需要与数据库做数据的交互，因此数据库的设计的完整及合理性就显得十分重要。

聊天信息表（text）是聊天室模块的核心，这张表中的字段分为两类，即访问者信息（昵称、发言时间、发言类型）和聊天信息（发言内容、发送表情、动作）。这两类字段就可以完整的描述一次发言的全部信息。下面来看具体的设计步骤。

（1）建立 chatroom 数据库，SQL 语句如下：

```
CREATE DATABASE 'chatroom ;
```

（2）在 chatroom 数据库中建立一个命名为 text 的聊天信息表，结构如表 10.1 所示。创建表及相关字段的 SQL 语句如下：

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
--
-- 数据库: 'chatroom'
--

-- --------------------------------------------------------

--
-- 表的结构 'text'
--

CREATE TABLE 'text' (
  'serial' int(5) NOT NULL auto_increment,
  'chatname' char(20) NOT NULL,
  'chattype' char(20) NOT NULL,
  'chattime' time NOT NULL,
  'chattext' text,
  'chatemote' char(30) default NULL,
  'chataction' char(30) default NULL,
  PRIMARY KEY  ('serial')
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=6 ;
```

表 10.1　聊天信息表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
serial	int	5	否	auto_increment	发言序列号（自增 ID）
chatname	char	20	否		发言者姓名
chattype	char	20	否		发言类型
chattime	time				发言时间 Current_time()
chattext	text		是		发言内容
chatemote	char	30	是		发言表情
chataction	char	30	是		发言动作

以上 text 表可以满足一个聊天室的基础需要，随着聊天室访问者的增多，可以考虑将单独的一张表分为 3 张表，即聊天序列表（chatsession）、详细信息表（message）、用户

信息表（userlist）。这样，程序可以按照业务功能获得数据库中指定的数据，可以有效地减少由于多个用户对同一张表操作所造成的锁表等问题。

（3）为了方便下一步的程序测试，向数据库的 text 表中插入初始化测试数据，创建语句如下：

```
--
-- 导出表中的数据 'text'
--

INSERT INTO 'text' VALUES (1, 'yachty', 'say', '22:15:32', 'hi all', '',
NULL);
INSERT INTO 'text' VALUES (2, 'muyizi', 'act', '22:16:13', '大家好', NULL, '双');
INSERT INTO 'text' VALUES (3, 'muyizi', 'say', '22:16:33', '大家好', '', NULL);
INSERT INTO 'text' VALUES (4, 'muyizi', 'act', '22:16:42', '', NULL, '双');
INSERT INTO 'text' VALUES (5, 'muyizi', 'act', '22:16:55', '', NULL, '快');
```

10.3　访问者引导

当一个浏览者进入聊天室，首先需要选择一个适合的主题，然后再进入对应的主题区开始讨论、交流等。访问者引导流程的核心是合理地引导访问者选择聊天主题，然后进入聊天室，当然这也是实现人员分流的一个设计（缓解流量过于集中的压力）。也有部分聊天室把这个页面作为新用户注册的页面进行，如设置昵称、用户权限加载等操作。具体的功能划分可以根据聊天室的具体情形选择设定。

10.3.1　引导页面设计

聊天室的引导页面也是聊天室的欢迎页面，在这个页面里访问者需要选择聊天的主题，并填写一些最基本的个人信息，比如名字、性别、年龄等。也可以加入注册个人信息的表单。在本例中，只要访问者填写昵称和选择聊天主题即可。创建 index.php 文件用来做引导页面，该文件的代码如下：

```
<title>聊天室引导页面</title>
</head>
<body>
<form action="index.php" method="POST" name="f1">
    <center>
        <h1>
            <font color=blue>欢迎进入 PHP 聊天室</font>
        </h1>
            <div>请填写昵称：
      <input type="text" name='nickname'></div>
            <div>请选择聊天主题：
            <select name="topic">
                <option value=""></option>
                <option value='金融聚焦'>金融聚焦</option>
                <option value='股市风云'>股市风云</option>
                <option value='保险基金'>保险基金</option>
                <option value='精英理财'>精英理财</option>
```

```
            <option value='都市情感'>都市情感</option>
        </select>
        <input type="submit" value="进入聊天室" /></div>
    </center>
    </form>
</body>
```

在浏览器中运行 index.php，效果如图 10.3 所示。

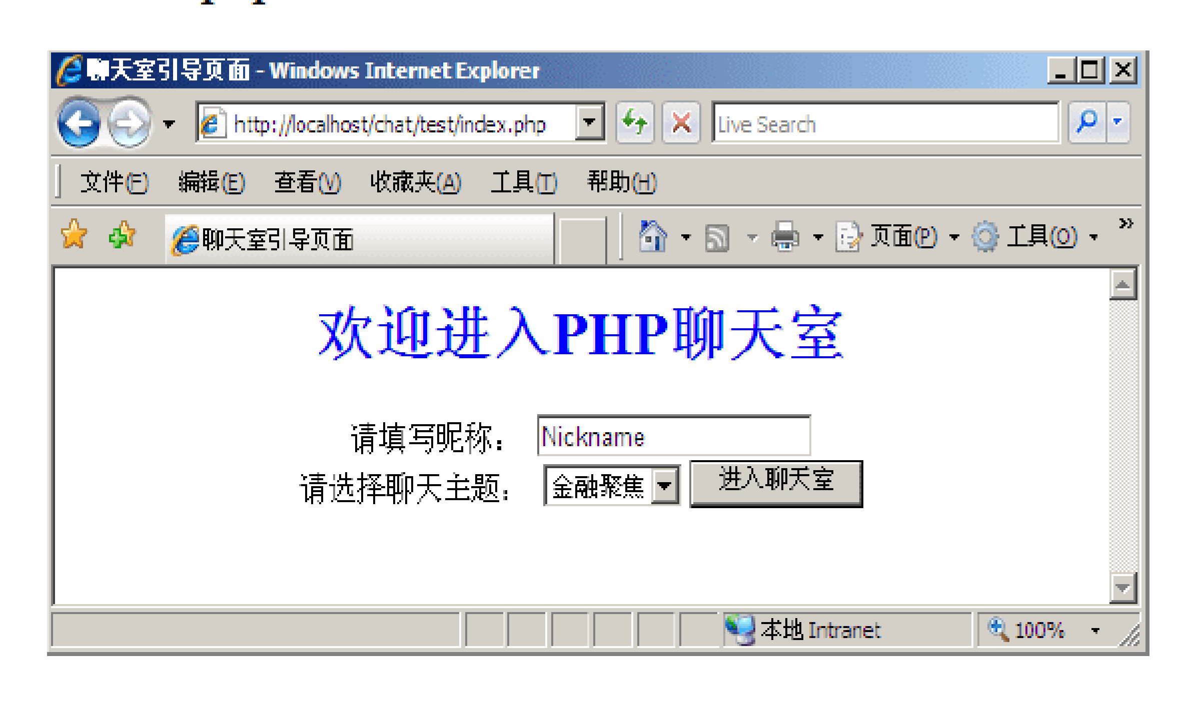


图 10.3　聊天引导页面效果图

【代码解读】

通过上面的代码可以看到，index.php 是单纯的 HTML 标签页面，最核心的部分是表单的设计。在表单的部分包含了 3 个标签，<input type="text" name='nickname'>用来输入用户昵称；select 下拉框用来选择当前要进入的主题；<input type="submit" value="进入聊天室"/>标签用来触发表单发送。

10.3.2　引导页面程序

根据系统的功能设计要求，引导页面程序需要完成 3 个主要的功能：

（1）创建两个（昵称和主题）Session 的全局变量以便后续的其他程序引用。

（2）判断录入信息的完整性和异常情况处理（包括 Session 变量的销毁），即流程控制。

（3）当选择完毕后跳转到聊天室主界面。

打开 index.php 文件，加入程序逻辑部分的代码，具体代码如下：

```
    <?php
    session_start ();
    if (! empty ( $_POST ['nickname'] ) and ! empty ( $_POST ['topic'] )) {
        //创建当前回话者全局变量
        $_SESSION ['user'] = $_POST ['nickname'];       //获得昵称变量
        $_SESSION ['topic'] = $_POST ['topic'];         //获得主题变量
        //var_dump ($_SESSION['user']);
        if (! empty ( $_SESSION ['user'] )) {
            header ( "Location: http://" . $_SERVER ['HTTP_HOST'] . rtrim
            ( dirname ( $_SERVER ['PHP_SELF'] ), '/\\' ) . "/" . "main.php" );
        } else {
            echo '<script type="text/javascript">alert(\'请正确输入昵称\')
            </script>';
            exit ();
        }
    } else {
```

```
        unset ( $_SESSION ['user'] );  //这种方法是将原来注册的某个变量销毁
        session_destroy ();            //这种方法是销毁整个 Session 文件
    }
    ?>
    <title>聊天室引导页面</title>
    </head>
    <body>
    <form action="index.php" method="POST" name="f1">
    <center>
    <h1><font color=blue>欢迎进入 PHP 聊天室</font></h1>
    <div>请填写昵称： <input type="text" name='nickname'></div>
    <div>请选择聊天主题：
<select name="topic">
        <option value=""></option>
        <option value='金融聚焦'>金融聚焦</option>
        <option value='股市风云'>股市风云</option>
        <option value='保险基金'>保险基金</option>
        <option value='精英理财'>精英理财</option>
        <option value='都市情感'>都市情感</option>
    </select>
<input type="submit" value="进入聊天室" />
</div>
    </center>
    </form>
```

【代码解读】

以上代码的执行步骤如下。

（1）通过 empty()函数判断表单传递过来数据的有效性，该函数先会确认传递过来昵称变量（$_POST['nickname'];）和主题变量（$_POST['topic'];）；

（2）创建存放昵称和当前主题的全局变量（$_SESSION['user']和$_SESSION['topic']）。

注意：session_start();语句必须放在文件的最上端。如果在它之前程序有输入则会造成 session 函数的报错，当 session 全局变量创建完毕后使用 header()函数跳转到聊天室框架页面（main.php）。相信读者对 header()函数并不陌生，但是在这里使用了 PHP 提供的系统变量来配合 header()函数完成跳转，这样程序的可移植性就更强了，不存在路径依赖。

10.4　聊天室程序

前一节介绍的是进入聊天室之前的引导页面，方便访问者选择话题和录入基本信息，并不涉及聊天室的业务部分。本节将逐步地介绍聊天室核心程序的设计和功能实现，按照完成的功能可以分为 4 个组成部分，即编辑发送信息（say.php）、显示全部信息（show.php）、聊天主题列表（list.php）和聊天室框架（main.php）。

10.4.1　编辑发送信息

编辑发送信息是聊天室模块的核心功能，该功能的好坏能直接地影响到用户的使用体

验。随着人们交流方式的多样，使用聊天室的用户已经不满足仅仅是“一问一答”的传统方式。因此，我们设计的这个聊天室模块中，包含了可以发送表情、动作的功能，这将极大的丰富用户的交流方式提升用户的使用体验。

1．编辑发送信息页面

编辑发送信息页面按照功能分区可以分为个人信息区（昵称）、发言区、动作选择区和表情选择区。核心的模板代码如下：

```
<body bgcolor="#0066FF">
<form action="say.php" method=post>
<table border=0 width=100%>
    <tr>
        <td width="64">昵称：</td>
        <td width="92"><input type=text name=chatuser size=8 value="昵称">
        </td>
        <td align=center width="32">说：</td>
        <td width="364"><input type=text name=chattext size=30
            maxlength=500></td>
        <!--发送语言或者动作信息-->
        <td><input type=submit value="送出"></td>
        <tr>
            <td><br>
            </td>
            <!--让网友选择发言类型是动作还是语言-->
            <td width="92"><input type="Radio" name=behavior value="say"
                checked></td>
            <td width="32">表情</td>
            <!--让网友选择发言的表情-->
            <td><select name=emote>
                <!--注意要保留一个表情选项为空，不是不带任何表情-->
                <option>
                <option>傻呆呆地
                <option>惊奇地
                <option>笑咪咪地
                <option>吞吞吐吐地
                <option>愤怒地
                <option>语重心长地
                <option>迷惑地
                <option>双手抱拳讨好地
            </select>
            <tr>
                <td><br>
                </td>
                <td width="92"><input type="Radio" name=behavior value="act">
                </td>
                <td width="32">动作</td>
                <td colspan="3"><select name=action>
                    <!--让网友选择聊天的动作-->
                    <option>双手抱拳，作个揖道：各位朋友请了！
                    <option>开始认真考虑
                    <option>挺起胸膛，大声喊道：让我来说！
                    <option>摇了摇头，叹道：还不明白
                    <option>板着脸，咬着牙说：不！我怎么这么笨！
                    <option>凄婉地说道：看来，我还得再看看书！
```

```
                    <option>大声叫嚷：你们都干吗呀？这么安静，怎么都没人和我聊天？
                    真没劲！
                    <option>快乐地唱道：我明白了！
                    <option>叹了口气，说道：还得努力！
                    <option>深深地叹了口气
                </select>
</table>
</form>
</body>
```

【代码解读】

以上代码是编辑发言区的页面模板代码，主要完成输入发言内容、表情、动作等功能，在技术细节上主要是 HTML 表单的展示。将上面的代码保存为 say.php 文件，在浏览器中运行该文件，效果如图 10.4 所示。

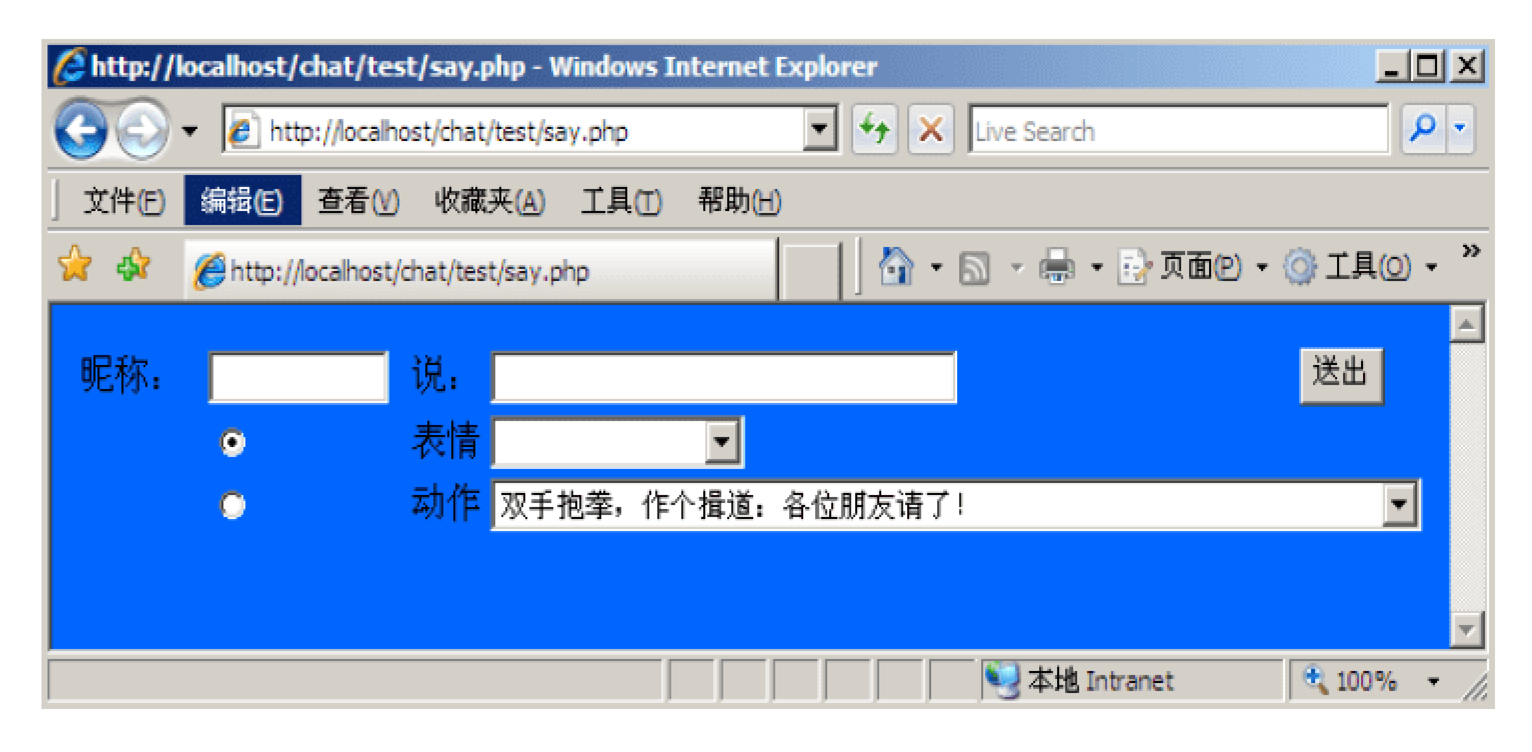


图 10.4　编辑发送信息页面

2．编辑发送信息程序

下面来看如何改造上面的页面文件，加入 PHP 程序脚本完成信息的编辑和发送。为了避免冗余，这里只列出增加的代码部分，嵌套后的完成代码读者可以参考源程序部分，完成信息编辑和发送加入两个程序段。

（1）程序段 1 的代码如下：

```
<?php
//功能：编辑发言程序 1（嵌套在页面模板最顶端）
session_start ();                //获得设定的 session 全局变量
var_dump ($_SESSION['user']);    //验证 user 变量
var_dump ($_SESSION['topic']);   //验证主题变量
?>
```

【代码解读】

以上代码初始化 Session 变量，检查之前设定的 Session 系统变量是否被正确地设置。

（2）程序段 2 放在页面模板的最底端，用来处理发送信息的服务器端逻辑，程序段 2 的代码如下：

```
<?
//保存发言信息或动作到信息数据库
//实现判断发言类型
if ($behavior == "say") {
    if (($chatuser != "") and ($chattext != "")) {
        //建立与 SQL 数据库的连接
```

```
            $connection = @mysql_connect ( "localhost", "root", "password" ) or
            die ( "无法连接数据库！" );
            @mysql_query ( "set names 'utf-8'" );
            @mysql_select_db ( "chatroom" ) or die ( "无法选择数据库！" );
            //插入发言信息
            $query = "INSERT INTO text(chatname,chattype,chattime, chatemote,
            chattext)";
            $query .= " VALUES ('$chatuser','$behavior',CURTIME(),'$emote',
            '$chattext')";
            $result = @mysql_query ( $query, $connection ) or die ( "存入数据
            库失败！" );
            mysql_close ( $connection ) or die ( "关闭数据库失败！" );
      }
}
if ($behavior == "act") {
      if ($chatuser != "") {
            //建立与 SQL 数据库的连接
            $connection = @mysql_connect ( "localhost", "root", "password" ) or
            die ( "无法连接数据库！" );
            @mysql_query ( "set names 'utf-8'" );
            @mysql_select_db ( "chatroom" ) or die ( "无法选择数据库！" );
            //插入发言信息
            $query = "INSERT INTO text (chatname,chattype,chattime,chattext,
            chataction)";
            $query = $query . " VALUES ('$chatuser','$behavior',CURTIME(),
            '$chattext','$action')";
            //echo $query;
            $result = @mysql_query ( $query, $connection ) or die ( "存入数据
            库失败！" );
            mysql_close ( $connection ) or die ( "关闭数据库失败！" );
      }
}
?>
```

【代码解读】

上面的程序会判断使用者发出留言信息或动作，程序就会判断聊天信息的类型，然后根据类型选择相应的字段插入，保存在 MySQL 数据库中。

使用单引号标识是为了在真正执行时，把实际数值代入到语句中。另外，语句中的变量 curren_time 是 MySQL 的常量，MySQL 在执行时会自动把当前的时间数据代入到该变量中。使用表格规划页面，并使用表单向网友提供动作或表情的选项，而单击“送出”提交按钮的处理程序是自身（say.php）。

```
<input type=text name=chatuser size=8 value="<?php echo $_SESSION
['user'];?>">
```

注意：在 HTML 代码中嵌入 PHP 脚本，由于在前面的引导程序已经将访问者注册的昵称作为全局变量保存在 Session 变量中，在文件顶端使用 session_start()初始化 session，然后程序通过读取 session 中的值，就可以获得全局昵称$_SESSION[‘user’]变量，使得使用者不必每次发言都要重新输入昵称。

10.4.2　显示全部信息

前面通过编辑发送消息程序，使用者已经可以编辑并发送自己的留言了。这个过程相

当于被编辑过的消息按照一定的规则，被插入到 chatroom 数据库的 text 表中了。现在创建名为 show.php 的 PHP 脚本文件，该脚本的功能是将使用者已经发送成功的信息，从数据库中遍历出来并按照一定的逻辑显示出来。下面来看具体的代码实现。

```
<?php
session_start ();
var_dump ($_SESSION['topic']);                    //验证 topic 变量是否正确设置
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>显示全部信息页面</title>
<!-- 每隔 10 秒就刷新一次本页面-->
<META http-equiv="Refresh" content="10;url=show.php">
<META content="text/HTML;charset=utf-8" http-equiv=Content-type>
<BODY>
<strong>
    <font color=blue>当前主题为：<?php echo $_SESSION ['topic'];?></font>
</strong>
<br>
<?php
//建立与 SQL 数据库的连接
$connection = @mysql_connect ( "localhost", "root", "password" ) or die ( "
无法连接数据库！" );
@mysql_query ( "set names 'utf-8'" );          //设置数据字符集
@mysql_select_db ( "chatroom" ) or die ( "无法选择数据库！" );
//向数据库发送查询请求
$query = "SELECT * FROM text ";
$result = @mysql_query ( $query, $connection ) or die ( "浏览失败！" );
//读取记录数据，分类保存
$count = 0;
while ( $row = mysql_fetch_array ( $result ) ) {
    //保存共有信息
    $gb [$count] [serial] = $row [serial];
    $gb [$count] [name] = $row [chatname];
    $gb [$count] [type] = $row [chattype];
    $gb [$count] [time] = $row [chattime];
    //根据发言类型，保存不同的信息
    if ($gb [$count] [type] == "say") {
        $gb [$count] [emote] = $row [chatemote];
        $gb [$count] [text] = $row [chattext];
    }
    if ($gb [$count] [type] == "act") {
        $gb [$count] [action] = $row [chataction];
    }
    $count ++;
}
mysql_close ( $connection ) or die ( "关闭数据库失败！"); //关闭数据库
//输出发言信息
for($i = $count - 1; $i >= 0; $i --)                    //循环遍历发言信息
{
    //根据发言类型输出相应的方法信息
    $chatstring = null;
```

```
    $chatstring .= $gb [$i] [time];
    $chatstring .= "<strong>  " . $gb [$i] [name] . " </strong>";
    //把各字段信息连接字符串输出
    if ($gb [$i] [type] == "say") {
        $chatstring .= "  ";
        $tempemote = $gb [$i] [emote];
        $chatstring .= "<font color=Maroon>  " . $tempemote . "</font>";
        $chatstring .= "说：";
        $chatstring .= $gb [$i] [text];
    }
    //判断发言类型
    if ($gb [$i] [type] == "act") {
        $chatstring .= "::";
        $chatstring = $chatstring . $gb [$i] [action];
    }
    echo "$chatstring<br>";
}
?>
</BODY>
</HTML>
```

通常 HTML 文件需要浏览者来选择浏览的内容，如果想自动更换显示的内容一般要用到 JavaScript、Java 或 CGI 等方法来完成。

在上面的程序中，使用 META 实现自动更换显示内容，更换的时间和文件都可以自行设定，语句如下：

```
<HEAD>
<TITLE>刷新内容</TITLE>
<META HTTP-EQUIV="REFRESH" CONTENT="x; URL=*.*">
</HEAD>
```

其中 x 是刷新的时间，单位是秒。*.*是刷新的文件。利用<META>标签实现 Web 的自动跳转，该标签直观、形象地描述是：

```
< meta http-equiv="Refresh" content="秒数; url=跳转的文件或地址" >
```

在显示全部信息程序中，下面的代码实现了页面每隔 10 秒刷新页面一次功能。

```
<META http-equiv="Refresh" content="10;url=show.php">
```

读取发言信息这段代码的程序运行流程如下：

（1）创建数据库服务器连接，打开数据库 chatroom，读取数据表 text 中的所有记录，存储在数据集$result 中，

（2）依次获取数据集$result 中的记录，将共有的信息记录到数组$gb 中。

（3）根据发言的类型，分别将相关信息记录到数组$gb 中。

输出发言信息使用变量$chatstring 来记录发言的内容，按照发言的类型选择存储在数组$gb 中的发言信息。由于最新发言的信息要置于页面的最上方，所以循环采用了逆序的方法。每行输出的格式有两种，如果是留言信息，就采用“时间+昵称+表情+说：留言信息”格式。如果是动作，就采用“时间+昵称+动作”格式，并对表情以区分颜色显示。

在浏览器中运行 show.php 程序，页面效果如图 10.5 所示。

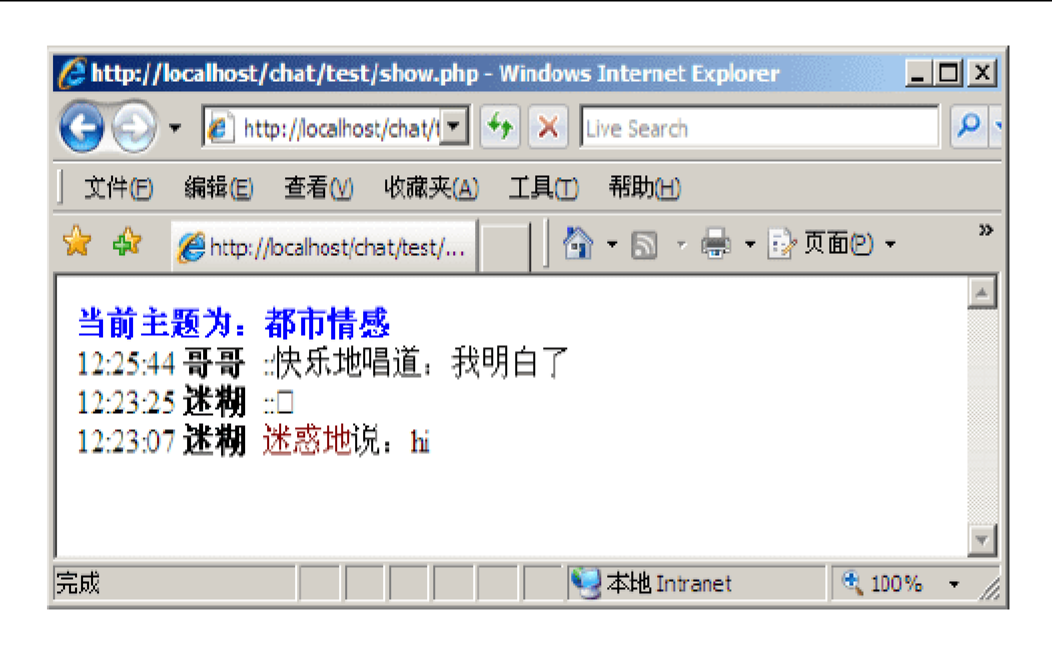


图 10.5　显示全部信息

10.4.3　聊天主题列表

聊天主题列表是用来给聊天室提供使用者栏目导航，通过栏目导航使用者可以在各个主题栏目之间切换，同时作为一个管理功能面板保留以便于以后新增功能的扩展。创建文件名为 list.php 的文件，具体的代码如下：

```
<?php
session start ();
//var dump ($ SESSION['user']);    //测试是否正确获得使用者
//var_dump ($_SESSION['topic']);   //测试是否正确获得主题
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>聊天主题列表</title>
</head>
<body>
<table id="mytable" cellspacing="0">
    <tr>
        <th scope="col" abbr="Configurations" class="nobg">金融聚焦</th>
    </tr>
    <tr>
        <th scope="row" abbr="Model" class="spec">股市风云</th>
    </tr>
    <tr>
        <th scope="row" abbr="G5 Processor" class="specalt">保险基金</th>
    </tr>
    <tr>
        <th scope="row" abbr="Frontside bus" class="spec">精英理财</th>
    </tr>
    <tr>
        <th scope="row" abbr="L2 Cache" class="specalt">都市情感</th>
    </tr>
</table>
</body>
</html>
```

在浏览器中运行 list.php，效果如图 10.6 所示。

10.4.4　聊天室结构框架

细心的读者可能注意到了，所有的程序和页面都是完成独立的功能，要想实现聊天的

功能需要在不同的页面和程序之间做切换，这种模式对用户十分不友好。为了解决这个问题，现在需要创建一个页面的框架，它规划一个聊天室的功能布局，并将不同的页面和程序加载进来。

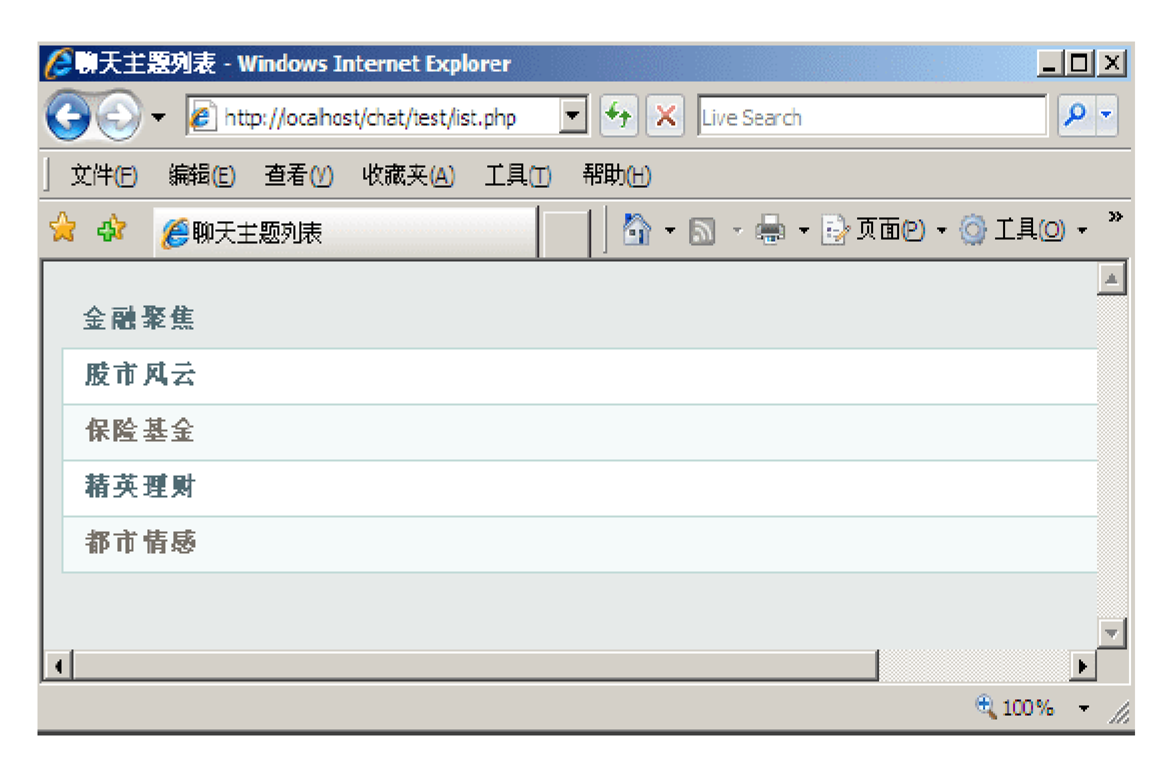


图 10.6　聊天主题列表

现在创建 main.php 文件并开始框架部分的设计。在 HTML 标签中，有一个 Frameset 框架的标记，主要是分割窗口和插入浮动窗口的功能。Frameset 框架重要属性如下所示。

- Rows：框架所占用的行数（rows="10,*"表示占 10 行，rows="*"表示占窗口全部行）。
- Cols：框架所占用的列数（cols="10,*"表示占 10 列，cols="*"表示占窗口全部列）。
- Framespacing：框架之间的距离。
- Frameborder：框架边框是否显示 frameborder="yes"或者 frameborder="no"。
- Border：框架边框的粗细。
- Frameset：标记定义分割窗口。
- Frame：标记定义窗口。

注意：Frameset 标记是个双标记。

Frameset 具体用法是先用 frameset 标记分割父窗口，把它分为两个子窗口，在用 frame 标记定义子窗口。

聊天室框架程序在 main.php 中，具体的代码如下：

```
<html>
<head>
<title>聊天室框架</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<FRAMESET rows="70%,*">
    <FRAMESET cols="80%,*">
        <frame name="view" src="show.php" noresize scrolling="no">
        <frame name="list" src="list.php" noresize scrolling="no">
    </FRAMESET>
    <frame name="send" src="say.php" noresize scrolling="no">
</FRAMESET>
<noframes>
</noframes>
</html>
```

在浏览器中运行 main.php 文件，效果如图 10.7 所示。从图中可以看到，之前调试好的各个子程序，都已经被正确地加载到框架中了。

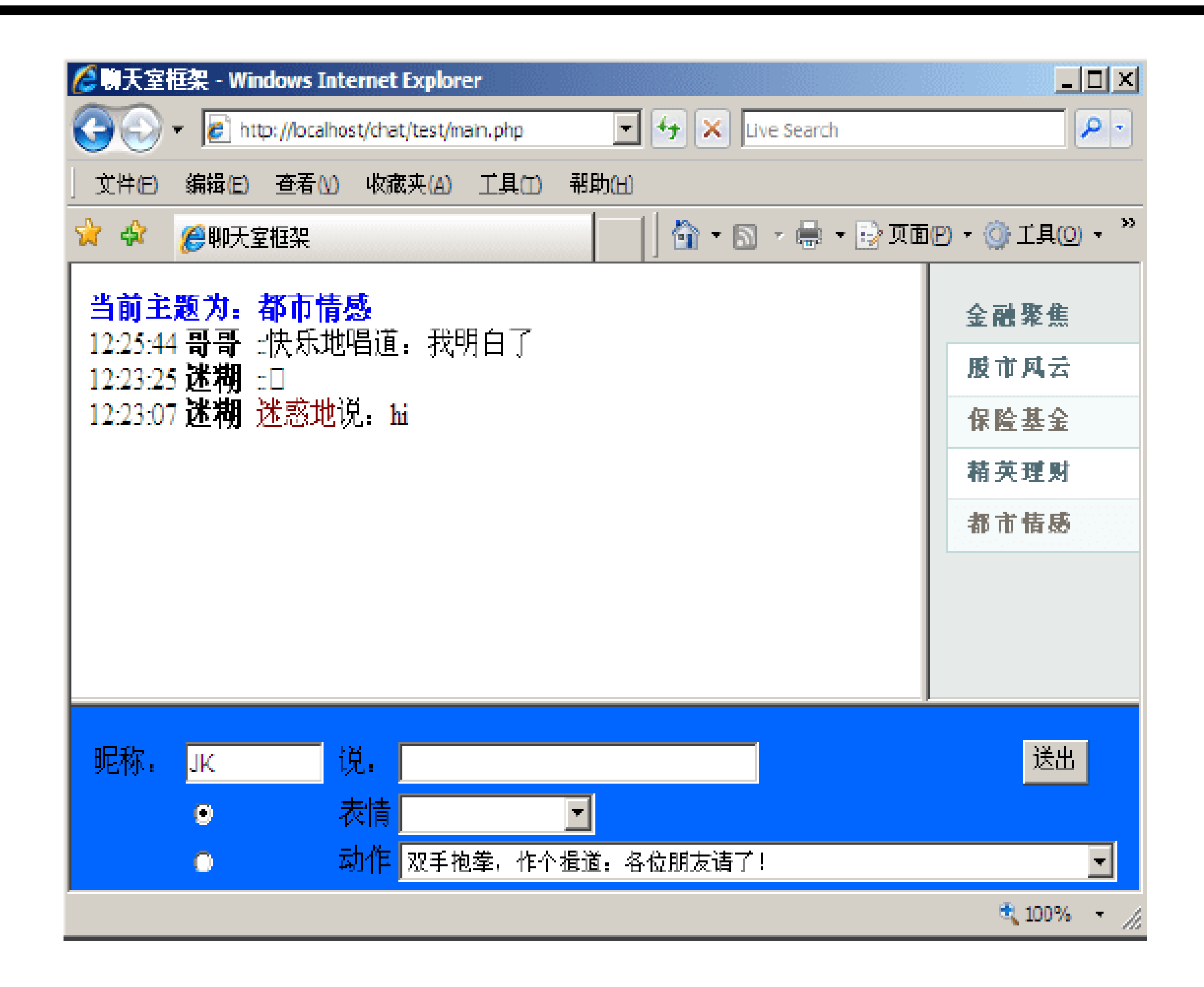


图 10.7　聊天室完整框架

10.5　聊天室的优化和完善

在前几节我们已经成功构建出了一个聊天室程序，但这个程序更多的是教给读者在框架程序下，如何整合不同的功能程序完成一个完整的应用。任何程序都需要不断地完善，读者可以根据自己的喜好，对这个聊天室模块做进一步地优化和改进。下面给出可以改进和添加功能的具体建议。

（1）发言信息的管理：分页显示，只是对最新页面的更新。当然最新页面总是位于第一页。

（2）各聊天主题的在线人数统计：统计各个主题聊天室在线人数和人员，并能密切关注人员的离开和进入。

（3）聊天方式：可以加入图片方式的表情等。

（4）聊天内容监控：聊天室的管理员界面，对不合主题的聊天内容进行删除，并对捣乱的网友取消其权限。这样也要建立聊天会员制度或聊天群制度。

10.6　小　　结

聊天室是一类比较常见的 Web 应用程序，这类程序的特点是程序与访问者有很强的互动性，这需要开发者对项目有整体规划和整合的能力。本章着重介绍聊天室的业务流程及技术实现原理，读者需要理解聊天室程序的逻辑结构设计。本章的技术要点是 PHP 中的服务器变量（Session）实际运用、交互式程序的实现方法等。

第 11 章　计费系统模块（原子操作+数据回滚 ROLLBACK）

网站的计费系统是网站的重要组成部分，该系统承担着网站计费相关的核心工作。本章将介绍计费系统模块的数据库设计、程序设计原则，以及创建不同类型的应用接口程序。本章引入了模板引擎程序，通过它实现 PHP 代码和页面的分离，并将模板引擎技术应用在充值记录程序之中。完整地学习完本章内容之后，读者会对虚拟货币相关的设计和程序开发有清楚的认识。

本章主要涉及的知识点如下。

- 原子操作：在执行完毕前不会被任何其他任务或事件中断。
- PHP 函数封装：将复用性高的代码段封装成公共函数，便于其他程序引用。
- Smarty 模板引擎：实现了将 HTML 代码与 PHP 代码逻辑分离。
- PHP 接口程序：PHP 接口程序是面向对象编程的一个具体实现。

11.1　计费系统概述

目前商业运营的网站都有独立的计费系统，这是商务运营的业务需要。同时，虚拟货币的存在，也为用户提供了更好的互动性，极大地提升服务质量，运营得当会使得网站与访问者形成良性的循环。

由于上面提到的原因，计费系统就变得十分重要。同时，由于涉及用户的付费行为，因此对系统的安全性要求是最高的。这里引入原子操作的概念，众所周知，原子是不可分割的，因此原子操作在执行完毕不会被任何其他任务或事件中断。在计费系统的设计中，使用原子操作的理念来实现系统的安全性和对异常情况的处理。这里笔者所说的原子操作包含以下两个方面。

（1）某个流程（如充值流程）需要包含几个步骤（如更新账户货币数量、记录这次充值的操作详情等），也就是说，只有当两个操作都完成了之后，这个流程才算完成（可以把这个流程作为一个不可分的原子），流程中任何一个环节出了问题，整个流程中已经开始的操作要回滚到初始的状态。

（2）原子操作常见的方法是通过数据库回滚来实现原子操作。具体的过程是使用 PHP 实现数据库回滚操作，下面的伪代码描述了回滚的流程：

```
<?php
//建立数据库连接
mysql_query ( 'BEGIN' );
$SQL = "...";
```

```
mysql_query ( $SQL );//做相应的数据库操作
//判断回滚条件:
if(mysql_errno){
   print   mysql_error();
   //异常情况数据回滚
   mysql_query('ROLLBACK');
   exit();
 }
//结束回滚操作
mysql_query('COMMIT');
?>
```

【代码解读】

上面的代码是实现数据回滚操作的标准流程，这个流程包含 3 个步骤：

（1）判断回滚条件（在 MySQL 数据库条件下可以判断 mysql_errno 状态）。

（2）异常情况数据回滚（在 MySQL 数据库条件使用 mysql_query('ROLLBACK');）

（3）结束回滚操作。

11.2　数据库设计规划

计费系统模块的数据库设计目的是，记录用户的虚拟货币数量，对用户消费和充值情况作统计。由于涉及网站的充值和消费功能，因此这部分的数据库设计就显得非常重要，在这里也采用了不同以往的表设计。计费系统数据库（billing）包含 3 个表，即用户虚拟货币存储表（user_account）、用户交易明细表（account_detail）、支付平台订单表（order）。下面将介绍具体的数据库设计。

11.2.1　用户虚拟货币存储表设计

用户虚拟货币存储表（user_account）的设计目的是统计用户账户的金额，它是计费系统的核心表。因为用户的充值和消费情况，最终就是通过这张表的记录反映出来的，涉及消费的业务逻辑也会预先检查账户的金额是否够支付。下面来看具体的设计步骤。

（1）建立 billing 数据库，SQL 语句如下：

```
CREATE DATABASE 'billing ;
```

（2）在 billing 数据库中建立一个命名为 user_accout 的用户虚拟货币存储表，结构如表 11.1 所示。创建表及相关字段的 SQL 语句如下：

```
--
-- 表的结构 'user_account'
--

CREATE TABLE 'user_account' (
  'id' int(11) NOT NULL auto_increment,
  'uid' int(11) NOT NULL,
  'increase' int(11) NOT NULL default '0',
  'decrease' int(11) NOT NULL default '0',
  'createtime' datetime NOT NULL,
```

```
  'lastupdate' datetime NOT NULL,
  'count' int(11) NOT NULL,
  PRIMARY KEY  ('id'),
  KEY 'idx_uid' ('uid')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

表 11.1　用户虚拟货币存储表

字　段	数据类型	长　度	NULL	默　认　值	字段说明
id	int	11	Not null	auto_increment	自增主键
uid	Int	11	Not null		用户 id
increase	Int	11	Not null	Default‘0’	增加总金额
decrease	int	11	Not null		减少总金额
createtime	datetime		Not null		账户创建时间
lastupdate	datetime		Not null		账号最后一次更新时间
count	int	11	Not null		操作次数

用户虚拟货币存储表中的字段设计跟以往记录类型的表有很大的不同，原因在于，这张表是记录用户货币信息的，它需要满足业务的需要具体解释：

（1）当查询某个用户的账户余额时，需要用 increase 字段中的值减去 decrease 字段中的值，即总的充值额减去总的消费额就是当前账户的余额。

（2）当用户充值时就对应在 increase 字段加入对应的充值数量，消费的情况也是一样的。lastupdate 字段记录该账号最后一次更新的时间，这样可以对长期不活跃的用户发邮件提醒或者是清除操作。每次对该表的操作（不管是充值或者是消费）都会使 count 字段做加 1 的操作，这样设计是为了发现异常的对货币的操作。

11.2.2　用户交易明细表

用户交易明细表（account_detail）的设计目的是记录用户的交易记录，包括充值和消费。它会详细地记录每笔交易的详细情况（时间、金额、交易状态、详细描述等），便于用户和网站管理人员查询。

在 billing 数据库中，建立一个命名为 accout_detail 的用户交易明细表，结构如表 11.2 所示。创建表及相关字段的 SQL 语句如下：

```
--
-- 表的结构 'account detail'
--

CREATE TABLE 'account detail' (
  'id' int(11) NOT NULL auto_increment,
  'uid' int(11) NOT NULL,
  'createtime' datetime NOT NULL,
  'transtype' tinyint(1) NOT NULL,
  'score' int(11) NOT NULL,
  'desctext' varchar(254) default NULL,
  'status' tinyint(4) NOT NULL,
  PRIMARY KEY  ('id'),
  KEY 'idx uid' ('uid')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

表 11.2　用户交易明细表

字　　段	数 据 类 型	长　　度	NULL	默　认　值	字 段 说 明
id	int	11	Not null	auto_increment	自增主键
uid	Int	11	Not null		用户 id
createtime	datetime		Not null		创建时间
transtype	tinyint	1	Not null		交易类型，0 为消费，1 为充值
score	int	11	Not null		金额
desctext	varchar	254	null		交易描述
status	tinyint	4	Not null		状态标志位，0 为未完成，1 为完成

说明：transtype 字段表示交易类型 0 表示消费，1 表示充值。具体的解释是，无论充值还是消费都对应的一条记录，需要根据这个标志位的值来确认该条记录是消费还是充值。Status 字段表示状态标志位，它是根据实际业务的需要设置的，如充值的流程有异常可能中断，这时该条记录已经创建了。只有当实际扣费完成后及状态标志位变成 1 之后，才能被认为该次充值或者消费已经生效了，这在实际的业务中是很有意义的。

11.2.3　支付平台订单表

支付平台订单表（order）是计费系统里的一张附表，它的设计目的是将多种支付手段单独剥离出来。这样便于为各个不同平台结算和付费统计，也使得系统更容易扩充。

在 billing 数据库中建立一个命名为 order 的支付平台订单表，结构如表 11.3 所示。创建表及相关字段的 SQL 语句如下：

```
--
-- 表的结构 'order'
--

CREATE TABLE 'order' (
  'id' int(11) NOT NULL auto increment,
  'a id' int(11) NOT NULL,
  'type' tinyint(1) NOT NULL,
  'order id' varchar(20) NOT NULL,
  PRIMARY KEY  ('id')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

表 11.3　支付平台订单表

字　　段	数 据 类 型	长　　度	NULL	默　认　值	字 段 说 明
id	int	11	Not null	auto_increment	自增主键
a_id	Int	11	Not null		用户交易表的流水 id
type	tinyint	1	Not null		交易平台类型：1 云网，2 快钱，3 支付宝，0 其他
order_id	varchar	20	Not null		订单 id

说明：支付平台订单表本身是作为计费系统中的一张附表，它单独使用并无意义，需要配合其他两张表一起使用，程序中的逻辑也是这样设计的。

11.3　核心功能函数

计费系统模块中很多基础的功能复用率很高，因此将这些核心的功能封装成公共函数库（include.functions.php），这样在完成具体的程序时，就不用过多地考虑程序功能方面的实现，只要关注逻辑和流程控制就可以了。本节是计费相关的核心函数说明。

11.3.1　使用 Charge_num()函数给指定用户充值

该函数的功能是给指定用户 uid 充值，add 为充入的具体数量（其中不包含 count+1 的操作）。Charge_num()函数代码如下：

```
//功能：给指定用户 uid 充值
//说明 add 为充入的具体数量
function Charge num($uid,$add,$mysql)
{
    $sql="UPDATE 'user account' SET 'increase' = 'increase'+".$add." ,
    'lastupdate'=now() WHERE 'uid' = '".$uid."' LIMIT 1 ;";
    $result = mysql query($sql, $mysql);
    if($result){
        return  true;
        }else{
        return false;
    }
}
```

【代码解读】

该函数具体的操作是对数据库中 user_account 表中的 increase 字段做加法操作。

11.3.2　使用 decrease_num()函数实现消费金额扣减

该函数的功能是扣减对应 uid 的货币数量（消费金额），$cdd 变量为扣减的数量（其中不包含 count+1 的操作）。decrease_num 函数代码如下：

```
//接口扣减对应 uid 的货币数量(消费金额)
//说明：默认相信 uid 是合法的
//结合 check uid 检查 uid 的有效性
function decrease_num($uid,$cdd,$mysql){
    //if(ereg("^[0-9\_]*$",$cdd))
    if(ereg("^[+]?[0-9]*\.?[0-9]+$",$cdd)){
$sql="UPDATE  'user_account'  SET  'decrease'  =  'decrease'+".$cdd.",
'lastupdate'=now()  WHERE 'uid' = '".$uid."' LIMIT 1 ;";
        $result = mysql query($sql, $mysql);
        if($result){
                $op='true';
        }else{
            $op='false';
        }
        return $op;
    }
}
```

【代码解读】

具体的操作是对数据库中 user_account 表中的 decrease 字段做加法的更新操作，并通过 now()函数更新操作时间。

11.3.3　使用 Transaction_record()函数记录用户的交易

该函数的功能是根据用户 uid 插入用户当次的交易记录，$record 为具体参数的数组，如果插入成功返回 true，如果失败返回 false。函数代码如下：

```
//根据用户 uid 插入用户的当次的交易记录
//$record 为具体参数的数组，如果插入成功返回 true，如果失败返回 false
function Transaction record($record,$mysql)
{
    $sql="INSERT INTO 'account detail' ( 'id' , 'uid' , 'createtime' ,
    'transtype' , 'score' , 'desctext' ) ";
    $sql.="VALUES (";
    $sql.="'','".$record['uid']."',now(),'".$record['transtype']."','".$
    record['score']."', '".$record['desctext']."'";
    $sql.=");";
    //echo $sql;
    $result = mysql query($sql, $mysql);
    if ($result)
    {
        $op='true';
    }else{
        $op='false';
    }
    return $op;
    }
```

【代码解读】

具体的操作是对数据库中 account_detail 表中插入一条数据，用来记录当次的交易。

11.3.4　使用 Update_status_record()函数确认用户付费操作

该函数的功能是确认用户付费成功了之后，将 account_detail 中的 status 设置为 1（1 表示充值成功，0 为失败）。函数代码如下：

```
//确认用户付费成功了之后将 account detail 中的 status 设置为 1 表示充值成功，0 为失败
//备注：需要加入订单号来确认唯一数据，之后再修改充值状态
function Update status record($cid,$mysql)
{
    $sql="SELECT * FROM 'account detail' WHERE id='".$cid."' LIMIT 1 ;";
    $result = mysql query($sql, $mysql);
    $row = mysql fetch array($result);
    if($row['status']=='0')
    {
        //暂取用户插入的时间最新的数据
        $sql="UPDATE 'account detail' SET 'status' = 1 WHERE 'id' = '".$cid."'
        LIMIT 1 ;";
        $result = mysql query($sql, $mysql);
        if ($result)
        {
```

```
            $op='true';
        }else{
            $op='false';
        }

    }else{
        $op='1';
        }
        return $op;
}
```

【代码解读】

具体的操作是，先从 accout_detail 表中判断是否有这条数据，然后加入订单号来确认唯一数据，之后再修改充值状态。

11.3.5　使用 count_user_num()函数更新充值次数

该函数的功能是用户做充值动作后，将 user_account 更新 count 的数值后加 1，记录某个用户一共充值的次数（包含未支付成功的）。函数代码如下：

```
//说明：对 uid 账号做统一的 count 加 1 的操作
//用户做充值动作后将 user_account 更新 count 的值数值加 1
//记录某个用户一共充值的次数(包含未成功的)用来和 account_detail 的记录条数匹配来确认数据是否异常
//$record 为具体参数的数组，如果插入成功返回 true，如果失败返回 false
//功能：仅仅增加记录交易次数，不对账户做充值动作
function count_user_num($uid,$mysql)
{
    $sql="UPDATE 'user_account' SET 'count'='count'+1 WHERE 'uid' = '".$uid."' LIMIT 1 ;";
    $result = mysql_query($sql, $mysql);
    if ($result)
    {
        $op='true';
    }else{
        $op='false';
    }
    return $op;
}
```

【代码解读】

具体的操作是用来和 account_detail 的记录条数匹配，来确认数据是否异常。$record 为具体参数的数组，如果插入成功返回 true，如果失败返回 false，仅仅增加记录交易次数，不对账户做充值操作。

11.4　应用程序接口

在前一节把对数据库中表操作的常用功能都封装成了功能函数，本节将利用这些函数创建程序内部调用接口程序，这些封装好的接口将完成计费系统的基本功能。其他扩展的程序可以根据接口中预先规定好的格式进行业务操作，不必也不允许直接对计费系统数据

库进行操作，这样可以有效地避免错误的产生，同时规范开发流程。

本节的 3 个接口程序分别使用了 3 种不同的接口返回类型，读者可以根据这 3 种不同类型接口的特点，在构建自己的应用程序时选择相应的接口程序类型。

11.4.1　充值接口程序

充值接口程序用来对指定的用户账户进行充值操作，先来定义充值接口程序的样式：

```
http://localhost/billing/value.php?uid=**&score=**&desctext=**
```

【代码解读】

其中，uid 为账户 id，如果是首次充值则为用户在 user_accont 表中创建一个账户、score 的值为充入的金额、desctext 为充值信息描述（如充值来源说明等）。

下面创建一个名为 value.php 的 PHP 脚本文件，该脚本程序的功能是通过用户 uid 为指定的账户充值，该脚本程序的代码如下：

```
<?php
 /*
    *功能：充值接口程序根据用户 uid 为指定用户充值
    *说明：充值成功后返回充值金额，异常情况则抛出报错
    */
    //数据库配置文件
    require once(dirname(  FILE  )."/inc/inc.read.dbconfig.php");
    //函数库文件
    require once(dirname(  FILE  )."/include/include.functions.php");
    $mysql   =   mysql connect($mysqlconf["host"],   $mysqlconf["user"],
    $mysqlconf["pass"]);
    mysql select db($mysqlconf["db"], $mysql);
    $uid=var_process('get','uid');              //获得 uid
    $score=var process('get','score');          //获得充值金额
    $desctext=var_process('get','desctext');    //获得充值描述
    //查询 uid 是否存在，不存在就在数据库中创建对应的查询账户
    $uid r=check uid($uid,$mysql);
    if($uid r=='true'){
        if($uid&&$score){
            $record['uid']=$uid;                //生成当次交易记录的数组
            $record['transtype']='1';           //交易类型 0 消费,1 充值
            $record['score']=$score;            //金额
            $record['desctext']=$desctext;      //交易描述
            //如果记录交易数据成功会返回 true，失败返回 false
            $Transaction record=Transaction record($record,$mysql);
            //对 account_detail 操作，记录当前交易记录
            if($Transaction record=='true'){
                //增减财富的操作
                $Charge num=Charge num($uid,$score,$mysql);
                //对 user_account 表进行操作,扣增加货币数量
                    if(!empty($Charge num)){
                            echo "充值成功,金额".$score;
                        }else{
                            echo "充值失败";
                    }
            }
        }else{
```

```
                echo "输入信息不全";
            }
}
?>
```

【代码解读】

由于使用了预先定义好的功能函数来完成业务逻辑，上面的代码变得十分简练，并且逻辑结构和流程控制更为清晰，这也是使用功能函数的重要意义。该充值程序的程序执行步骤如下：

（1）对传入的 3 个参数使用 var_process()安全接收，防止注入攻击及因传入非法数据造成的异常（如负数等）；

（2）使用 check_uid()函数检查 uid，即货币账户的信息，如果没有 uid 则创建一个货币账户；

（3）通过 Transaction_record()和 Charge_num()函数完成记录操作记录和对指定账户充值的操作。

执行下面的链接：

```
http://localhost/billing/value.php?uid=10&score=50&desctext=测试
```

浏览器中返回信息如下：

```
充值成功,金额 50
```

11.4.2　扣费接口程序

扣费接口程序用来对指定的用户消费进行扣费，先来看定义充值接口程序的样式：

```
http://localhost/billing/decrease.php?uid=**&score=**&desctext=**
```

【代码解读】

其中，uid 为账户 id，如果 uid 不存在则抛出报错，score 的值为当次扣费的金额，desctext 为扣费信息描述（比如消费说明等）。

该程序与充值接口程序有所不同的是，扣费接口程序对操作中的异常和操作结果的返回值都预先做了定义（详见代码注释）。这样，其他的开发人员在新的程序中使用该接口时，可以根据接口文档来调试自己的程序。这样做的好处是，起始程序异常也不会将错误的内容暴露出去，提高程序的安全性。

下面创建一个名为 decrease.php 的 PHP 脚本文件，该程序的功能是扣除指定 uid 账户的金额，该脚本程序的代码如下：

```
<?php
    //功能:扣减用户 uid 对应的账户的金额
    //根据用户 uid 扣除当次消费的货币
 /**
     * 根据用户 uid 查询用户余额
     * @uid:用户身份 id
     *
     * @返回:
     * -1:增减财富的操作失败
```

```
     * -2:输入信息不全
     * -3:uid 创建出错
     * -4:输入非法字符，没有获得 uid 的值
     * -0:空
     * 0:成功
     *
     */
    require_once(dirname(__FILE__)."/inc/inc.read.dbconfig.php");
                                                    //数据库配置文件
    require_once(dirname(__FILE__)."/include/include.functions.php");
                                                    //函数库文件
    //创建数据库链接
    $mysql = mysql_connect($mysqlconf["host"], $mysqlconf["user"],
    $mysqlconf["pass"]);
    mysql_select_db($mysqlconf["db"], $mysql);     //选择相应的数据库
    $uid=var_process('get','uid');                  //获得 uid
    $score=var_process('get','score');              //获得扣费金额
    //扣费描述
    $desctext=var_process('get','desctext');
    if($uid&&$score&&$desctext){
        $record['uid']=$uid;                        //生成当次交易记录的数组
        $record['transtype']='0';                   //交易类型 0 消费，1 充值
        $record['score']=$score;                    //金额
        $record['desctext']=$desctext;              //交易描述
        //如果记录交易数据成功会返回 true，失败返回 false
        $Transaction_record=Transaction_record($record,$mysql);
                                    //对 account_detail 操作，记录当前交易记录
        if($Transaction_record=='true'){
            //增减财富的操作
            $decrease=decrease_num($uid,$score,$mysql);
                                    //对 user_account 表进行操作，扣减货币数量
            if($decrease=='true'){
                    //记录账户操作的次数，便于核对
                    $count=count_user_num($uid,$mysql);
                    echo "0\n" .$score;
                }else{
                echo "-1\n";
            }
        }
    }else{
    echo "-2\n";
}
?>
```

【代码解读】

扣费接口程序的逻辑与充值接口程序非常相似，该程序的执行步骤如下：

（1）对传入的 3 个参数使用 var_process()安全接收，防止注入攻击及因传入非法数据造成的异常（如负数等）；

（2）使用 Transaction_record()函数向数据库中插入一条数据，用于记录本次操作；

（3）使用 decrease_num()函数对“user_account”表进行操作，扣减货币数量，再通过$count=count_user_num()函数记录账户操作的次数，该操作便于后续的账户核对。

执行下面的链接：

```
http://localhost/billing/decrease..php?uid=10&score=30&desctext=测试
```

浏览器中返回信息如下：

```
0 30
```

通过预先定义的接口返回值可以知道扣费成功，扣费金额为 30。

11.4.3　用户余额查询程序

根据用户的 uid 查询账户余额是计费系统模块中最常用的接口，许多的操作都需要预先获得该用户的账户余额，这样才能避免因为余额不足导致的错误。由于用户余额查询的接口使用更为广泛，所以该程序将扣费的结果信息通过标准的 XML 格式提交给浏览器，作为其他程序调用的标准数据容器。

创建一个名为 balance_xml.php 的 PHP 脚本文件，该脚本程序会根据用户 uid 查询用户余额并将结果生成 XML 文档。该脚本程序的代码如下：

```
<?php
    /**
     * 根据用户 uid 查询用户余额
     * 并将结果生成 XML 格式以便调用
     * @uid:用户身份 id
     *
     * @返回:
     * -1:uid 不存在
     * -2:数据库连接错误
     * -3:uid 创建出错
     * -4:输入非法字符，没有获得 uid 的值
     * -0:空
     * 0:成功
     *
     */
    require once(dirname(  FILE  )."/inc/inc.read.dbconfig.php");
                                                    //数据库配置文件
    require once(dirname(  FILE  )."/include/include.functions.php");
                                                    //函数库文件
    //需要安全性的判断比如:ipcheck
    /*
        if (!ipcheck())
    {
        //echo "-8\n"; //ip 认证错误
        exit;
    }
    */
    $uids=var_process('get','uid');                     //获得当前用户的 uid
    $mysql = mysql connect($mysqlconf["host"], $mysqlconf["user"], $mysqlconf
    ["pass"]);
    if (!empty($mysql)){
        mysql select db($mysqlconf["db"], $mysql);
        }else{
        mysql close($mysql);
        die("-2\n");
    }
//判断 uid 是否存在
if($uids){
```

```
    //判断输入的 uid 是否为纯数字
    if(ereg("^[0-9\_]*$",$uids)){
        $Surplus=Surplus($uids,$mysql);            //当前用户剩余的货币数量
        //var dump ($Surplus);
        //echo "uid".$uid."--剩余".$Surplus;
        //echo "0\n".$Surplus;
        //序列化数组，返回正确值
        }else{
        //echo "-1\n"; //uid 不存在
        exit;
    }
}
//生成 XML 文档
    echo " <xml id=\"tt\">";
    echo "<?xml version=\"1.0\" encoding=\"gb2312\"?>";
    echo "<ttl>".$Surplus."</ttl>";
    echo "</xml>";
?>
```

【代码解读】

上面的代码逻辑流程相对前面更为简单，在业务处理的部分，用 Surplus()函数来获得当前用户剩余的货币数量。其他逻辑判断的部分跟前面两个接口程序相同，这里不再重复。

在浏览器中执行下面的链接：

```
http://localhost/billing/balance_xml.php?uid=10
```

预览效果如图 11.1 所示，从图中可以看到该账户的余额为 270。

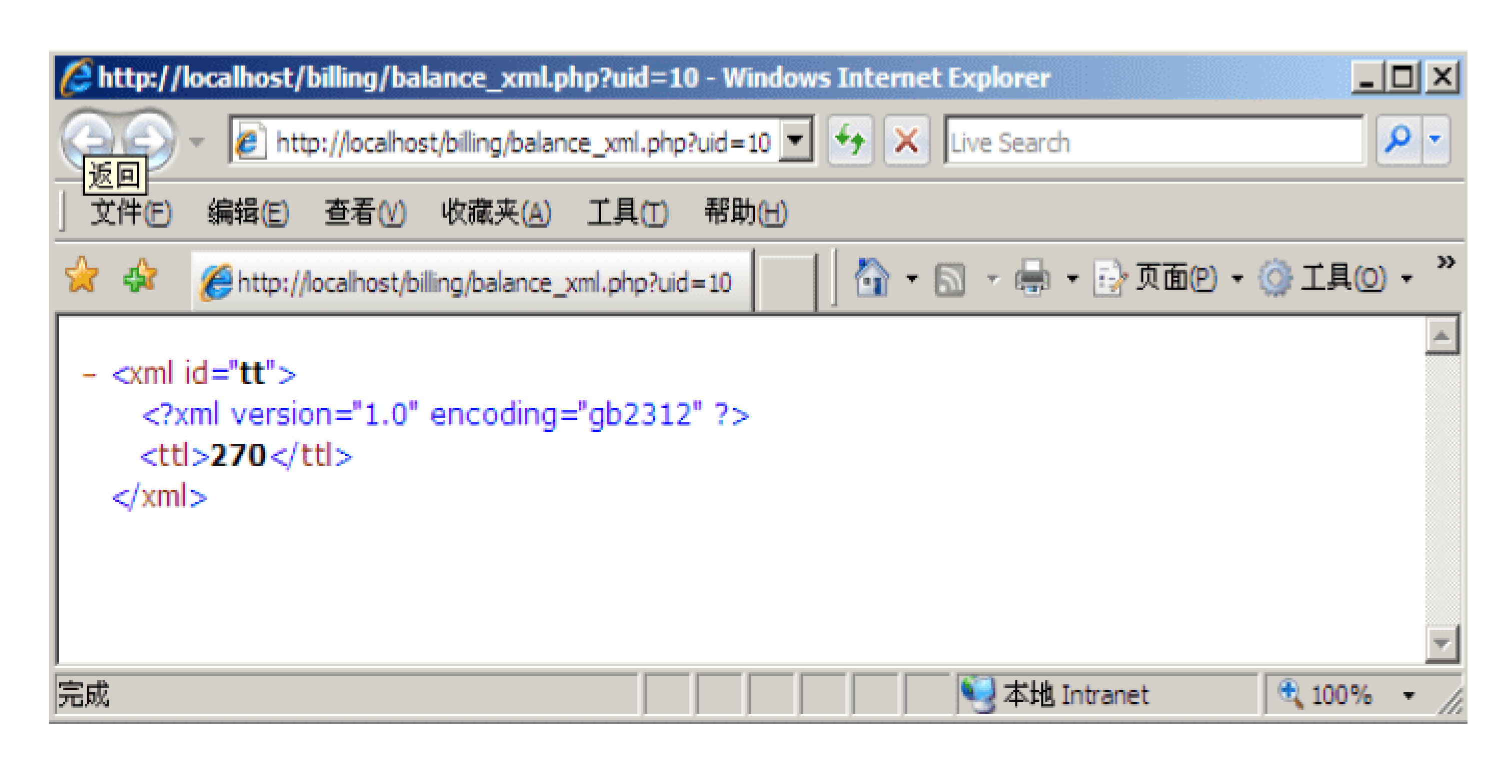


图 11.1　用户余额查询

11.5　Smarty 模板引擎

Smarty 是使用 PHP 语言编写的模板引擎，是目前被广泛使用的模板引擎之一。它分离了逻辑代码和外在的内容，提供了一种易于管理和使用的方法，用来将原本与 HTML 代码混杂在一起 PHP 代码逻辑分离。目的就是要使 PHP 程序同前端页面设计分离，使程序员改变程序的逻辑内容不会影响到前端人员的页面设计，前端人员重新修改页面不会影响到程序的程序逻辑，这在多人合作的项目中显得尤为重要。

Smarty 模板引擎的使用步骤如下。

（1）Smarty 的安装十分简单，首先从 www.smarty.net 站点上下载最新的 Smarty 版本，

解压缩后将其中的 libs 目录复制到服务器的 Web 根目录下，并且将文件夹改名为 Smarty。

（2）在 Smarty 目录下的 Smarty.class.php，是使用 Smarty 模板引擎进行开发必须包含的主文件。

下面在网站根目录下创建一个名为 test 的子目录，在 test 子目录下创建文件名为 Smarty_inc.php 的模板引擎配置文件，该文件的内容如下：

```
<?php
include_once ("../Smarty/Smarty.class.php");   //包含smarty模板引擎的核心文件
$smarty = new Smarty ();
//下面设置 smarty 模板引擎的运行环境
$smarty->caching = false;           //在开发阶段不使用缓存功能，所以设置为 false
$smarty->template_dir = "./templates";         //设置模板文件存放位置
$smarty->compile_dir = "./templates_c";        //设置编译后的 php 文件存放位置
$smarty->cache_dir = "./smarty_cache";         //设置缓存目录
?>
```

【代码解读】

上面代码是配置了 Smarty 模板引擎的初始化的参数配置，读者在使用的过程中可以参考代码的注释，了解变量的含义。

（3）在 test 目录下创建 index.php 文件，该文件是执行程序的主文件，用来完成业务逻辑并加载模板文件。该文件的内容如下：

```
<?php
include ("Smarty_inc.php");
//下面定义变量
$title = "网站标题！";
$content = "网站正文！";
//下面设置模板变量
$smarty->assign ( "title", $title );      //"title"是在模板文件中定义的变量名
$smarty->assign ( "content", $content ); //"content"是在模板文件中定义的变量名
$smarty->display ( "index.tpl" );    //显示模板内容，smarty 会用上面设置的变量值替
                                      换模板的中的变量。模板 index.tpl 在下面创建
?>
```

（4）在 test.php 脚本中，已经生成了 Smarty 的模板标签。接下来创建模板文件，在 test/templates 目录下创建文件 index.tpl，内容如下：

```
<html>
<head>
    <title>{$title}</title>
</head>
<body>
{$content}
</body>
</html>
```

（5）在浏览器中运行可看到页面的显示效果，如图 11.2 所示，这就是使用 Smarty 模板引擎的一个基础应用。

浏览 test/templates_c 目录查看生成的文件，可以看到一个很长的文件名的 PHP 程序，这是由 Smarty 模板引擎自动生成的编译过的 PHP 程序。

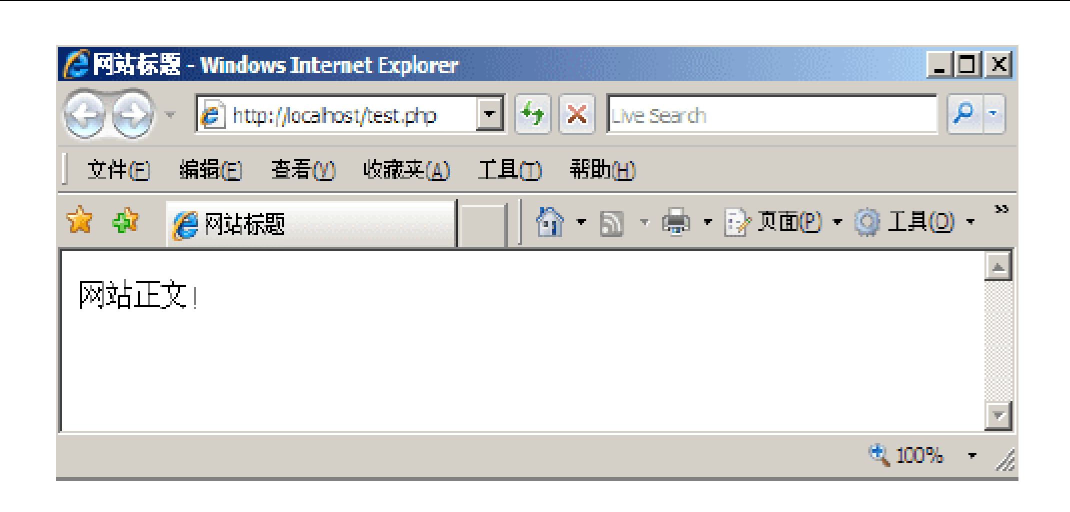


图 11.2　模板引擎基本应用

注意：Smarty 中的变量分割符号为大括号{}，这可能和 JavaScript 中的标签存在冲突，解决方法是使用$smarty->left_delimiter 和$smarty->right_delimiter 进行重新定义新的分割符号。例如可以使用如下定义：

```
$smarty->left_delimiter = "{|";
$smarty->right_delimiter = "|}";
```

11.6　查询充值记录

查询充值记录功能是提供给用户查询自己历史充值记录的操作界面，通过这个界面用户可以清楚地了解到自己的充值情况，也可以在认为自己账户有异常的时候，方便地查询和比对，查询充值记录是计费类站点的必要功能之一。

11.6.1　查询充值记录界面

查询充值记录界面按照功能可以分为信息显示列表、其他操作功能面板和系统消息提示 3 个部分。

创建一个名为 value_record.html 的查询页面，核心界面的模板代码如下：

```
<body>
<div id="Housecenter">
<div id="HouseGiveto">
<div class="Waretop">
    <span><img src="img/wptopr.gif" /></span>
          <img src="img/wptopl.gif" />
</div>
<div class="Waretopd">
  <div class="Waretopdw">我要充值-我的充值记录</div>
  </div>
  <div class="Warecnt HouseGiveIn">
<div class="shopLeftlistz Business">
    <ul>
        <li class="shopLeftcolor">
            <span>订单号</span>
            <span>交易日期</span>
            <span>消费额</span>
            <span>订单明细</span>
            <span>订单状态</span>
        </li>
    <li>
```

```
        <span>5</span>
        <span>2010-06-17 15:07:58</span>
        <span>50 金粉笔</span>
        <span>支付成功</span>
        <span class="Leftlistz">[成功]</span>
    </li>
    <li>
        <span>6</span>
        <span>2010-06-17 15:07:58</span>
        <span>50 金粉笔</span>
        <span>支付成功</span>
        <span class="Leftlistz">[成功]</span>
    </li>
    <!--省略，读者可参照程序源代码-->
</ul>
</div>
<form action="value record.php?uid=10&page=1" method="get">
<div id="footer">
输入页数 <INPUT onchange="" size=3 ame=pageno id ="page" name="page">
第 1 页, 共 2 页
| <A href="value record.php?uid=10&page=1" target= self>首页</A>
| <a href="value_record.php?uid=10&page=1">上一页</a>
| <A href="value_record.php?uid=10&page=2" target=_self>下一页</A>
| <A href="value record.php?uid=10&page=2" target= self>尾页</A>
</div>
</form>
<div class="RushvalueNote">
    <a href="pay.php">
        <img src="img/housecz.gif" border="0" />
    </a>
    <a href="Exchange.php" target=" blank"><img src="img/houseczb.gif"
    border="0" /></a>
</div>
<div class="LeftAsse">
<div class="LeftAssetop">
    <span><img src="img/glR.gif" /></span>
    <img src="img/glL.gif" width="34" height="24" />
</div>
<div class="LeftAssecnt">
<div class="HouseAsseText">
<div class="HouseAsseBack">
  <blockquote>
    <p><span class="HouseSmall">1 元人民币= <img src="img/wpfbj.gif" width=
    "5" height="14" /> 1<a href="#">金</a>粉笔</span>温馨提示：</p>
    <p><span class="HouseBig">1 个<a href="#">金</a>粉笔= <img src="img/
    wpfb.gif" width="5" height="14" /> 100 个白粉笔</span>此行为单向不可逆。</p>
  </blockquote>
</div>
<div class="HouseAsseFall">
  <input type="submit" name="Submit" value="【区别是？】" class="SuccesB" />
  <input type="submit" name="Submit2" value="【如何赚取？】" class="SuccesB"
/>
</div>
<div class="HouseAsseFallz"></div>
</div>
</div>
<div class="LeftAssebottom">
    <span><img src="img/glbR.gif" /></span>
```

```
            <img src="img/glbL.gif" width="56" height="11" />
</div>
</div>
  </div>
  </div>
</div>
```

【代码解读】

以上页面模板按照功能可以分为两个功能区域：

- 充值记录查询区域，该区域用来循环显示用户的充值记录。
- 操作面板区域，该区域用来扩展用户其他操作。

上述代码的设计要点是表单中内容标签的设计，读者可以通过这部分内容复习关于标签及表单的内容。

11.6.2　查询充值记录程序

查询充值记录程序要完成的功能有两个，一个是根据 uid 用预先定义的 Value_Inquiry() 函数，将充值记录数据从数据库中遍历出来生成 Smarty 的变量；另一个是处理翻页的数据并生成 Smarty 的翻页数组。

创建一个名为 value_record.php 的 PHP 脚本，该文件具体的代码如下：

```
<?php
//功能：充值记录查询
//1.根据用户 uid 查询用户消费明细
//2.提示充值
require_once (dirname ( __FILE__ ) . "/inc/inc.read.dbconfig.php");
                                                        //数据库配置文件
require_once (dirname ( __FILE__ ) . "/include/include.functions.php");
                                                        //函数库文件
require_once (dirname ( __FILE__ ) . "/include.smarty.php"); //正式的 smarty
$mysql = mysql_connect ( $mysqlconf ["host"], $mysqlconf ["user"],
$mysqlconf ["pass"] );
mysql_select_db ( $mysqlconf ["db"], $mysql );
$uids = $_GET [uid];                               //获得用户 uid
$check = check_uid ( $uids, $mysql );              //用户检查，不存在创建账号
if ($check == 'true') {
    $smarty->assign ( "uid", $uids );
    $month = var_process ( 'get', 'month' );
    $page = var_process ( 'get', 'page', 1 );           //默认值 0
    $smarty->assign ( "page", $page );                  //翻页函数
    //$no==$cur_no ? $no=$residual : $no=$nums;         //三元算子
    $sum_article = total_check ( $uids, $mysql );
    $offset = ($page - 1) * 5;                          //偏移值
    $nums = '5';                                        //单页显示条数
    $sum = ceil ( $sum_article / $nums );               //算出总页数
    $smarty->assign ( "sum", $sum );
    $month = "0";
    $Inquiry = Value_Inquiry ( $uids, $month, $offset, $nums, $mysql );
                                                        //充值明细查询
    $smarty->assign ( "Inquiry", $Inquiry );
    $divpage = divpage_toadmin ( $sum_article, $page, $nums );//翻页函数
    $smarty->assign ( "divpage", $divpage );                //翻页函数
```

```
    $smarty->display ( "./value_record.html" );              //正式需要做修改
}
//关闭数据库连接
mysql close ( $mysql );
?>
```

【代码解读】

上面的程序把用户充值记录信息和翻页数组生成 Smarty 变量，并输出到 value_record.html 页面中。现在打开 value_record.html 文件，将 HTML 标签替换成 Smarty 标签，并插入 Smarty 的变量循环标签。修改后的 value_record.html 代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>我的充值记录</title>
<meta name="keywords" content="我的充值记录" />
<meta name="description" content="我的充值记录！" />
<link href="css/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="Housecenter">
<div id="HouseGiveto">
<div class="Waretop">
    <span><img src="img/wptopr.gif" /></span>
        <img src="img/wptopl.gif" />
</div>
<div class="Waretopd">
  <div class="Waretopdw">我要充值-我的充值记录</div>
  </div>
  <div class="Warecnt HouseGiveIn">
<div class="shopLeftlistz Business">
    <ul>
        <li class="shopLeftcolor">
            <span>订单号</span>
            <span>交易日期</span>
            <span>消费额</span>
            <span>订单明细</span>
            <span>订单状态</span>
        </li>
{|section name=Inquiry loop=$Inquiry|}
    <li>
        <span>{|$Inquiry[Inquiry].id|}</span>
        <span>{|$Inquiry[Inquiry].createtime|}</span>
        <span>{|$Inquiry[Inquiry].score|} 金粉笔</span>
{|if $Inquiry[Inquiry].status=='失败'|}
        <span><a href="detail.php?cid={|$Inquiry[Inquiry].id|}">点此查看
        </a></span>
    {|else|}
        <span>支付成功</span>
{|/if|}
        <span class="Leftlistz">[{|$Inquiry[Inquiry].status|}]</span>
    </li>
{|/section|}
</ul>
</div>
<form action="value record.php?uid={|$uid|}&page={|$page|}" method="get">
<div id="footer">
```

```
输入页数 <INPUT onchange="" size=3 ame=pageno id ="page" name="page">
第{|$divpage.currpage|}页, 共 {|$sum|} 页
|<A href="value_record.php?uid={|$uid|}&page={|$divpage.firstpage|}" target=
_self>首页</A>
| <a href="value_record.php?uid={|$uid|}&page={|$divpage.prepage|}">上一
页</a>
| <A href="value_record.php?uid={|$uid|}&page={|$divpage.nextpage|}" target=
_self>下一页</A>
| <A href="value_record.php?uid={|$uid|}&page={|$divpage.max_page|}" target=
_self>尾页</A>
</div>
</form>
<div class="RushvalueNote">
    <a href="pay.php">
        <img src="img/housecz.gif" border="0" />
    </a>
    <a  href="Exchange.php"  target="_blank"><img  src="img/houseczb.gif"
    border="0" /></a>
</div>
<div class="LeftAsse">
<div class="LeftAssetop">
    <span><img src="img/glR.gif" /></span>
    <img src="img/glL.gif" width="34" height="24" />
</div>
<div class="LeftAssecnt">
<div class="HouseAsseText">
<div class="HouseAsseBack">
  <blockquote>
    <p><span class="HouseSmall">1 元人民币= <img src="img/wpfbj.gif" width="5"
    height="14" /> 1<a href="#">金</a>粉笔</span>温馨提示：</p>
    <p><span class="HouseBig">1 个<a href="#">金</a>粉笔= <img src="img/
    wpfb.gif" width="5" height="14" /> 100 个白粉笔</span>此行为单向不可逆。</p>
  </blockquote>
</div>
<div class="HouseAsseFall">
  <input type="submit" name="Submit" value="【区别是？】" class="SuccesB" />
  <input type="submit" name="Submit2" value="【如何赚取？】" class="SuccesB" />
</div>
<div class="HouseAsseFallz"></div>

</div>
</div>
<div class="LeftAssebottom">
    <span><img src="img/glbR.gif" /></span>
          <img src="img/glbL.gif" width="56" height="11" />
</div>
</div>
  </div>
  </div>
</div>
</body>
</html>
```

在浏览器中运行下面的链接：

```
http://localhost/billing/value_record.php?uid=10&page=1
```

查询充值记录程序运行的效果，如图 11.3 所示。

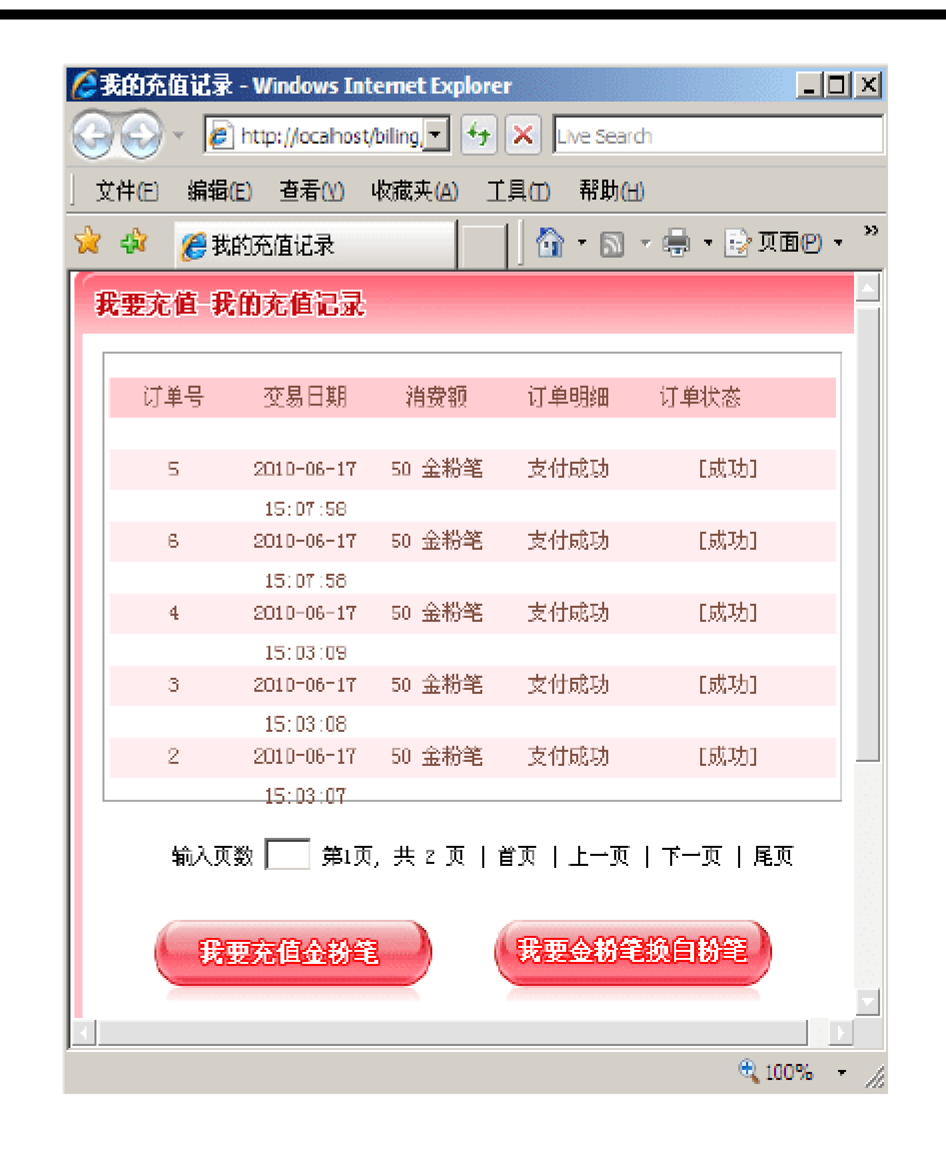


图 11.3　查询充值记录

11.7　小　　结

本章首先分析了网站计费系统，使读者对计费系统有个整体的了解，然后根据需求分析完成数据库部分设计、建立了数据库之后基于数据整理创建核心功能函数，核心功能函数包含了计费系统最常用的功能。在第 11.4 节将前面的核心功能函数封装成 3 个不同类型的接口程序，每个接口程序都完成了一个典型的功能应用。第 11.5 节引入了模板引擎的概念，它的引入实现了 PHP 代码和页面代码的分离，改变了传统 Web 开发的方式，因此有很强的现实意义。第 11.6 节实际应用了 Smarty 模板引擎的代码组织方式，通过这个方式完成了查询充值记录程序编程。

第 12 章　充值平台模块（MD5 加密 +HTTPS 协议）

随着电子商务的发展日渐成熟，各种类型的电子商务网站也如雨后春笋般出现。电子商务网站是为消费者提供网络电子消费服务的网站，其中核心的服务就是电子交易服务。本章通过第三方的支付接口建立一套完整的支付体系，集成各个银行的网上银行接口，提供给网站一个统一的充值平台，该平台的设计目的是方便用户完成与充值相关的基本操作、提供友好的界面引导。通过本章对充值平台模块程序的介绍，读者可以了解到如何设计针对第三方支付接口的交互程序，并了解平台的充值程序编程的基本方法。

本章主要涉及的知识点如下。

- PHP 中的 MD5 加密：PHP 脚本中最常用的字符串加密算法。
- 服务器间通知：指网关接口异步的调用用户预选设置的返回程序地址。
- HTTPS 协议：使用最广泛的安全超文本传输协议。
- mysql_insert_id()函数：返回给定的 link_identifier 中上一步 INSERT 查询中产生的 AUTO_INCREMENT 的 ID 号。

12.1　充值平台概述

充值平台程序是电子商务网站为消费者提供电子交易服务的平台，为了方便消费者使用该平台，集成了多家银行的网上银行接口。该平台设计的核心是针对第三方支付网关的编程，通过充值的流程了解如何编写与第三方支付网关交互的程序。

12.1.1　充值平台实现目的

充值平台的目的是通过第三方支付网关系统实现对网站用户的充值，充值平台设计的重点是通过充值的流程，了解通用的一个与接口程序交互的一般方法。因此，在数据库方面沿用“计费系统模块”的数据库（billing）设计，并没有再单独设计。

第三方的支付网关系统定义了一个和商户系统的开放接口，商户系统通过这一接口就可以使用众多现有流行的银行支付系统，完成网上交易，不必针对每家银行的支付系统进行设计。同时，系统还提供了完善的商家后台管理功能。

12.1.2　开发调试流程概述

在着手进入开发调试之前，首先需要在第三方支付公司的官方网站（https://www.cncard.

net）注册一个新用户，当申请的服务开通后，将会获得唯一的商户账号（包括商户号码、商户名及登录密码）。在网站页面左侧的“登录/申请区”输入商务号码、商户名、登录密码及随机码，单击“登录”按钮后即可进入商户管理后台。

也可以在浏览器中直接输入地址（https://www.cncard.net/cnpayment/admin/），进入登录商户管理后台，在管理首页获得商户的“支付密钥”，该密钥是 MD5 加密。该密钥最初生成是在商户申请完成时由系统自动生成，在商户管理后台可以修改。

在开始充值接口程序开发之前，首先需要对支付网关提供的接口程序进行联网调试，调试通过之后就可以进行编写“支付和充值”程序。网关支付完整流程如图 12.1 所示。

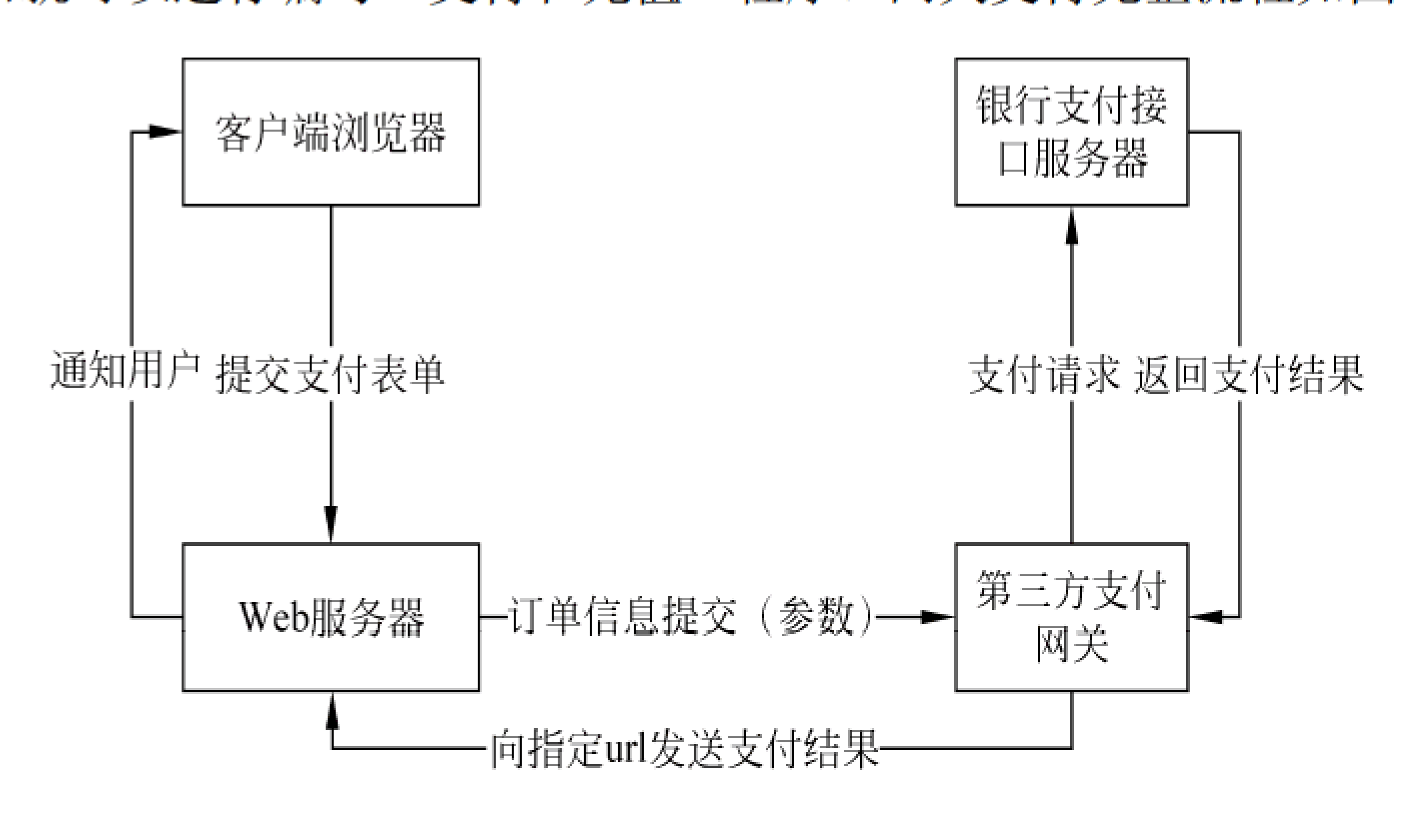


图 12.1　网关支付完整流程

具体的开发和调试流程如下。

（1）开发充值引导程序：主要的功能是创建与充值相关的表单（用户 id、订单号、充值金额等）。

（2）订单确认程序：主要功能是处理用户提交的信息，并生成支付网关要求的标准表单（准确传递支付网关参数）。

（3）接收网关支付信息程序：主要功能是接收网关支付信息、验证信息有效性、判断支付结果。

（4）充值程序：主要功能是接收支付网关支付通知信息、验证信息有效性、给用户显示支付结果和相关订单信息等。

12.2　测试接口说明

测试接口是把整个在线支付接口流程拆分为 4 个步骤，分别是签名测试模块、下订单测试模块、支付成功提货测试模块和支付网关整体测试模块。下面采用逐步开发测试的方法，这样可以提高开发效率、减少开发技术接口的时间。

12.2.1　签名测试模块

【测试目的】测试商户端的 MD5 加密结果和网关接口进行 MD5 加密结果是否一致。

【测试地址】https://www.cncard.com/purchase/cnpaygate/testport/port_test31.asp

【传递参数】源串（testStr）、MD5 加密结果串（signStr）。

【参数说明】

（1）testStr：未进行 MD5 加密前的一个字符串。

（2）signStr：对 testStr 串进行 MD5 加密后的结果串，提交到测试路径时，先后顺序可变。

例如：

```
https://www.cncard.com/purchase/cnpaygate/testport/port test31.asp?test
Str=aderutyigh&signStr=fe2aa1087c441ce9d8b4b3a378795532
```

12.2.2　下订单测试模块

【测试目的】测试商户端的订单信息是否能被“云网”网关系统正确接收。

【测试路径】https://www.cncard.com/purchase/cnpaygate/testport/port_test11.asp

【测试说明】商户端需要提供以下信息：c_mid、c_order、c_name、c_address、c_tel、c_post、c_orderamount、c_ymd、c_moneytype、c_retflag、c_returl、c_signstr、c_memo1、c_memo2、paygate、c_email（各项具体含义详见下文）。其中，c_signstr 是采用 MD5 对以下字符串：c_mid、c_order、c_orderamount、c_ymd、c_moneytype、c_retflag、c_returl、c_paygate、c_memo1、c_memo2 和 c_pass 进行加密得到加密串，如 c_signstr=MD5(c_mid & c_order & c_orderamount & c_ymd & c_moneytype & c_retflag & c_returl & c_paygate & c_memo1 & c_memo2 & c_pass)。

注意：按照上述先后顺序组成一个无间隔字符串进行 MD5 加密。其中，c_pass 为商户的支付密钥，云网支付网关正确接收到信息后，在页面显示出所接收的各项信息及验签结果。

12.2.3　支付成功提货测试模块

【测试目的】根据商户端提交的订单信息模拟测试结果是否正确。

【测试路径】https://www.cncard.com/purchase/cnpaygate/testport/port_test21.asp

【测试说明】首先由客户端访问上面提供的页面，然后根据页面要求填写各项信息并确认，“云网”服务器支付网关会根据提交的信息返回支付结果。

12.2.4　支付网关整体测试流程模块

【测试目的】在能正确完成上面三部分测试模块后，就可以模拟测试完整支付流程了。

【测试路径】https://www.cncard.com/purchase/cnpaygate/testport/getorder_test.asp

【测试说明】在客户端准备好需提交的各项信息后，访问上面提供的测试网页，并传递准备好的信息，经过支付网关端验证后进入如下页面。

```
https://www.cncard.com/purchase/cnpaygate/testport/createorder_test.asp
```

选择支付方式后进入以下链接：

```
https://www.cncard.com/purchase/cnpaygate/testport/handle_test.asp
```

浏览器加载以上链接后会在页面显示出支付结果。

12.3　通知支付结果方式的实现

当用户通过网上银行完成了支付流程之后，需要为用户返回“充值成功”页面。通常这个页面会包含本次的充值详情，还会有其他的操作提示，比如查看充值记录、继续购买等。

为了解决这个问题，要求网站支付程序接口必须提供一个支付结果通知命令。第三方支付网关在收到订单，确认页面发出的支付命令后，先执行支付操作，然后调用网站支付接口程序通知用户支付结果，同时取得网站支付程序接口提供的支付结果页面地址（由支付结果通知命令生成）。最后，支付网关系统把支付结果通知命令生成的支付结果页，返回用户的浏览器。

12.3.1　向支付网关提交支付请求信息

网站的充值程序需要向支付网关接口提交支付请求信息，网关接口地址如下：

```
https://www.cncard.net/purchase/getorder.asp
```

注意：为了保证交易的安全性，需要在提交时使用 HTTPS 协议（SSL）。

首先将 c_mid、c_order、c_orderamount、c_ymd、c_moneytype、c_retflag、c_returl、c_paygate、c_memo1、c_memo2、notifytype、c_language 和 c_pass 拼成一个字符串（其中 c_pass 是为商户的支付密钥，在商户申请成功后，登录商户管理后台 https://www.cncard.net/admin/后在首页可以得到该密钥，该密钥可以随时由商户修改），然后对于这个字符串再进行 MD5 加密，例如：

```
c_signstr =md5(c_mid & c_order & c_orderamount & c_ymd & c_moneytype &
c_retflag & c_returl & c_paygate & c_memo1 & c_memo2 & notifytype & c_language
& c_pass)
```

支付请求信息的参数信息详见表 12.1。

表 12.1　支付请求信息参数表

参数名称	参数含义	参数类型	长度	是否可为空	备　注
c_mid	商户编号	字符型	7	否	申请成功后获得的编号
c_order	订单号	字符型	20	否	网站生成的订单号
c_name	收货人姓名	字符型	10	可	中英文均可
c_address	收货人地址	字符型	200	可	中英文均可
c_tel	收货人电话	字符型	20	可	
c_post	收货人邮编	字符型	6	可	
c_email	收货人 Email	字符型	50	可	
c_orderamount	订单总金额	货币型		否	以元为单位，小数点后保留两位，如 13.45

续表

参数名称	参数含义	参数类型	长度	是否可为空	备　　注
c_ymd	订单产生日期	字符型	8	否	格式为“yyyymmdd” 如 20040301 或 20041218 注：月和日小于 10 的需要补 0
c_moneytype	支付币种	字符型	1	否	对应值固定为 0，代表人民币
c_retflag	返回标识	字符型	1	否	0—不用返回 1—返回到 c_returl 对应的 URL
c_returl	支付结果页面 URL	字符型		c_retflag 为 1 时不能为空，否则可为空	消费者完成购物后返回的商户页面 URL，c_retflag=1 时值不能为空，本地址需要完整填写文件（如 http://www.cncard.net/default.asp）且该文件里不能含有页面跳转的操作或语句
c_memo1	商户参数一	字符型	200	可	需要在支付结果通知中转发的参数一，不能包含中文
c_memo2	商户参数二	字符型	200	可	需要在支付结果通知中转发的参数二，不能包含中文
notifytype	支付结果通知方式	字符型	1	可	0 为普通通知方式，1 为服务器通知方式，空值为普通通知方式
c_signstr	交易信息签名	字符型	32	否	对订单数据进行 MD5 签名后的字符串
c_language	订单语种	整型	1	可	0 为中文，1 为英文，空为中文
c_paygate	支付方式	整型	4	可	当程序定义该值时，系统会自动转入银行的支付页面

上面的支付请求信息的参数信息表，详细地介绍了支付请求过程中涉及参数的含义。订单信息确认页面提交的表单样式如下：

```
<form name="form1" method="post" action = "http://www.cncard.net/purchase/getorder.asp ">
    <input type="hidden" name="c_mid" value="0000001">
    <input type="hidden" name="c_order" value="10001">
    <input type="hidden" name="c_name" value="张三">
    <input type="hidden" name="c_address" value=北京市朝阳区">
    <input type="hidden" name="c_tel" value="12345678">
    <input type="hidden" name="c_post" value="100001">
    <input type="hidden" name="c_email" value="**@vip.com">
    <input type="hidden" name="c_orderamount" value="13.45">
    <input type="hidden" name="c_ymd" value="20020503">
    <input type="hidden" name="c_moneytype" value="0">
    <input type="hidden" name="c_retflag" value="1">
    <input type="hidden" name="c_returl" value="http://www.cncard.com/purchase/result.asp">
    <input type="hidden" name=" c_paygate " value="1">
    <input type="hidden" name="c_memo1" value="0001">
    <input type="hidden" name="c_memo2" value="10001">
    <input type="hidden" name="c_language" value="0">
    <input type="hidden" name=" notifytype " value="0">
    <input type="hidden" name="c_signstr" value="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx">
    <tr>
    <td>订单号<td>10001
    <tr>
```

```
    <td>金　额<td>13.45
    <tr>
    <td>日　期<td>2002 年 5 月 3 日
    <tr>
    <td><input type=submit value="提交">
    </table>
    </form>
```

【代码解读】

以上的代码是支付请求表单模板代码，其中< input type >标签用来传送支付表单的参数，将标签的类型设置为“hidden”，该类型表示隐藏传送参数内容。读者在使用过程中，注意理解该标签的类型设置。

12.3.2　支付网关将支付结果信息传给程序的接口

网关采用 Get 方法，将参数按照预定顺序发给商户返回的 URL 地址，传回的参数名称和顺序，如表 12.2 所示。

表 12.2　支付网关传回的参数

参 数 名 称	参 数 含 义	参数类型	长度	是否可为空	备　注
c_mid	商户编号	字符型	7	否	申请成功后获得的编号
c_order	订单编号	字符型	20	否	生成的订单号，是由商户传递给网关的
c_orderamount	订单总金额	货币型		否	以元为单位，小数点后保留两位
c_ymd	订单产生日期	字符型	8	否	格式为“yyyymmdd” 如 20040301 或 20041218 注：月和日小于 10 的需要补 0
c_moneytype	支付币种	字符型	1	否	0 人民币
c_transnum	交易流水号	字符型	50	否	支付网关交易流水号（由网关系统产生）
c_succmark	交易成功标志	字符型	1	否	Y-成功 N-失败
c_cause	失败原因	字符型	100	可	支付失败时为失败原因
c_memo1	商户参数一	字符型	200	可	需要在支付结果通知中转发的商户参数一，不能包含中文
c_memo2	商户参数二	字符型	200	可	需要在支付结果通知中转发的商户参数二，不能包含中文
c_signstr	交易信息签名	字符型	32	否	商户对订单数据进行 MD5 签名后的字符串

以上表格中，c_signstr（交易信息签名）得到的方法是首先将 c_mid、c_order、c_orderamount、c_ymd、c_transnum 、c_succmark 、c_moneytype、c_memo1、c_memo2 和 c_pass 拼成一个字符串。

注意：其中 c_pass 是为商户的支付密钥，在商户申请成功后，登录商户管理后台 https://www.cncard.net/admin/后可以得到该密钥，而且商户可随时修改该密钥。

对于这个字符串再进行 MD5 加密，例如：

```
c_signstr = md5(c_mid & c_order & c_orderamount & c_ymd & c_transnum &
c_succmark & c_moneytype & c_memo1 & c_memo2 & c_pass)
```

12.3.3　服务器间通知方式

支付网关采用 Get 方法将支付结果参数发给商户提供的 c_returl 地址。在商户提供的 c_returl 页面中，商户需要根据接口的说明验证后，按以下格式输出：

```
<result>result</result><reURL>reURL</reURL>
```

【参数说明】

- result：值固定为 1，表示商户已成功收到网关的支付成功的通知。
- reURL：商户显示给用户处理结果页面的 URL。

处理结果页面 URL 样式如下：

```
<result>1</result><reURL> http://localhost/payment/GetPayHandle.php</reURL>
```

支付网关获取商户的通知反馈结果后，使用 Get 方法将支付结果参数提交给商户返回的 reURL 参数所对应的网址，并将商户的处理结果显示给用户。

如果因网络等原因，造成支付网关没有获取到商户对支付通知的反馈结果，将按以下机制向商户网关发送支付成功的通知：

（1）第 1 次通知失败后，网关按 1 秒、2 秒、4 秒、8 秒的间隔向商户发送通知，间隔时间倍增。

（2）当发送的间隔时间超过 30 分钟之后，每隔 30 分钟发送一次通知。第 1 次通知失败 24 小时之后，网关还无法获得网站的反馈结果，将对此订单停止发送通知。

12.4　充 值 引 导

在实际的充值业务中，用户首先会被引导到一个充值的页面。该页面将显示用户账户的余额信息，同时要求用户填写充值金额。在技术层面上，这实际上完成了创建用户、创建支付表单的过程，上述步骤是用户充值前在程序上的必要准备工作。

12.4.1　充值引导页面设计

在“充值引导”页面的设计中，提供给用户 3 个支付平台选择（第 2、3 个没有给出链接）。因为现在运营类的网站的充值方式很多，这种多样化的充值方式为不同的用户充值提供了充值便利，所以这里设计 3 个支付选择表单，可以方便读者今后使用这个页面扩展其他的充值渠道。创建文件名为 pay.html 的充值引导页面，由于该页面模板中包含 3 种充值方式，因此创建 JavaScript 函数用来处理页面显示逻辑代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>充值平台</title>
<meta name="keywords" content="充值平台" />
<meta name="description" content="充值平台" />
```

```
<link href="css/style.css" rel="stylesheet" type="text/css" />
<script>
        //显示充值方式 1 函数
          function meinv01(){
        //当调用 meinv01()函数时指定的 id 属性值得到对象，设置图片名称
          document.getElementById("pic01").src = "img/qhsj.gif";
          document.getElementById("pic02").src = "img/qhyh1.gif";
          document.getElementById("pic03").src = "img/qhxn1.gif";
        //当调用 meinv01()函数时指定的 id 属性值得到对象，设置对象状态为“隐藏”
          document.getElementById("dis01").style.display = "block";
          document.getElementById("dis02").style.display = "none";
          document.getElementById("dis03").style.display = "none";
          }
        //显示充值方式 2 函数
           function meinv02(){
        //当调用 meinv02()函数时指定的 id 属性值得到对象，设置图片名称
          document.getElementById("pic01").src = "img/qhsj1.gif";
          document.getElementById("pic02").src = "img/qhyh.gif";
          document.getElementById("pic03").src = "img/qhxn1.gif";
        //当调用 meinv02()函数时指定的 id 属性值得到对象，设置对象状态为“隐藏”
          document.getElementById("dis01").style.display = "none";
          document.getElementById("dis02").style.display = "block";
          document.getElementById("dis03").style.display = "none";
          }
        //显示充值方式 3 函数
           function meinv03(){
        //当调用 meinv03()函数时指定的 id 属性值得到对象，设置图片名称
          document.getElementById("pic01").src = "img/qhsj1.gif";
          document.getElementById("pic02").src = "img/qhyh1.gif";
          document.getElementById("pic03").src = "img/qhxn.gif";
        //当调用 meinv03()函数时指定的 id 属性值得到对象，设置对象状态为“隐藏”
          document.getElementById("dis01").style.display = "none";
          document.getElementById("dis02").style.display = "none";
          document.getElementById("dis03").style.display = "block";
          }
          </script>
</head>
<body>
```

【代码解读】

以上代码是控制页面支付方式显示逻辑的，具体的实现方法是根据调用不同的函数设置不同的图片名称，然后再将其他对象设置为“隐藏”，这样就完成了支付方式的显示功能。该方法在处理页面前端逻辑时经常用到，读者在使用时注意体会。

下面来介绍支付方式模板页面的设计。该模板由 3 个功能区组成，即账户余额显示区、支付方式选择区和操作流程提示区，下面具体介绍这几部分的代码实现。

账户余额显示区的页面代码如下：

```
<div id="Housecenter">
    <div id="HouseGiveto">
        <div class="Waretop">
            <span><img src="img/wptopr.gif" /></span>
            <img src="img/wptopl.gif" />
        </div>
<div class="Waretopd">
    <div class="Waretopdw">我要充值</div>
        <a href="value_record.php?uid=10">查看我的充值记录</a>
```

```
    </div>
    <div class="HouseGivetop">您的账户余额是:
        <img src="img/wpfbj.gif" width="5" height="14" />
{|$Surplus|}
<span>金</span>粉笔
    </div>
```

【代码解读】

以上代码用来显示当前用户的账户余额，这里定义模板标签{|$Surplus|}用来显示账户余额，由服务器端脚本程序返回具体的数值。

支付方式选择区的页面代码如下：

```
<div class="Warecnt myRushGive">
    <div class="myRusBorder">
        <div class="LeftAssetop">
        <span><img src="img/glR.gif" /></span>
            <img src="img/glL.gif" width="34" height="24" />
        </div>
<div class="myRushEdge">
<div class="myRushExc">
<a href="#"><img src="img/qhsj.gif" width="134" height="35" border="0"
id="pic01" onclick="meinv01()"/></a><a href="#"><img src="img/qhyh1.gif"
width="136" height="35" border="0" id="pic02" onclick="meinv02()" /></a><a
href="#"><img src="img/qhxn1.gif" width="133" height="35" border="0"
id="pic03" onclick="meinv03()"/></a>
</div>
<div class="myRushExcC" id="dis01">
    <form action="./PortSample/SendOrder.php?uid={|$uid|}" method="post">
        <div class="myRushExcI">
            <input type="text" name="textfield" class="" id="123"/></div>
            <span>输入本次充值的金粉笔数量</span>
            <input name="orderid" type="hidden" value="{|$orderid|}">
            <input type="submit" name="Submit2" value="【下一步】" class=
            "LeftTexttwo" />
    </form>
<div class="myRushExcF">
    <p>欢迎使用<strong>云网</strong>支付平台充值</p>
</div>
      <div class="myRushExcbot">提示：下一步您将跳转到银行网站，通过银行卡支付进
      行充值您所填写的资料全部都记录在银行，请您放心填写。</div>
</div>

<div class="myRushExcC" style="display:none;" id="dis02">
    <form action="" method="post" target="_blank" >
        <div class="myRushExcI">
            <input type="text" name="textfield" class="" id="123"/></div>
                <span>输入本次充值的金粉笔数量</span>
            <input name="orderid" type="hidden" value="{|$orderid|}">
            <input type="submit" name="Submit2" value="【下一步】" class=
            "LeftTexttwo" />
    </form>
<div class="myRushExcF">
    <p>欢迎使用<strong>支付宝</strong>支付平台充值</p>
</div>
      <div class="myRushExcbot">提示：下一步您将跳转到支付宝网站，通过你的支付宝
      账户进行充值，您所填写的资料全部都记录在支付宝，请您放心填写</div>
</div>
<div class="myRushExcC" style="display:none;" id="dis03">
```

```
    <form action="" method="post" target=" blank" >
      <div class="myRushExcI">
        <input type="text" name="textfield" class="" id="123"/></div>
            <span>输入本次充值的金粉笔数量</span>
        <input name="orderid"  type="hidden" value="{|$orderid|}">
        <input type="submit" name="Submit2" value="【下一步】" class="Left
        Texttwo" />
    </form>
<div class="myRushExcF">
    <p>欢迎使用<strong>神州行（快钱）</strong>支付平台充值</p>
</div>
<div class="myRushExcbot">提示：下一步您将跳转到快钱网站，通过快钱银行卡支付进行充
值，您所填写的资料全部都记录在快钱网站，请您放心填写。
特别提醒，进入快钱支付页面请选择“银行卡支付”，目前粉丝网仅支持此充值方式
</div>
</div>
<div class="myRushExcB"></div>
</div>
<div class="LeftAssebottom">
<span><img src="img/glbR.gif" /></span>
<img src="img/glbL.gif" width="56" height="11" />
</div>
</div>
<div class="myRusBorder">
<P><img src="img/myt.gif" /></P>
</div>
```

【代码解读】

以上页面代码是提供用户选择支付方式，处理选择流程的 JavaScript 代码在前面已经介绍过了，这里需要注意下面的标签：

```
<input name="orderid"  type="hidden" value="{|$orderid|}">
```

在用户选择支付方式时，页面模板会获得在前一个流程提交过来的该次充值的订单号，这里使用前面介绍过的 input 标签传送订单号数值。

操作流程提示区的代码如下：

```
<div class="myRusBorder">
<div class="myRushExp1">
<p><strong>充值流程</strong> <br />
    <img width="329" height="43" src="pay clip image001.jpg" /> </p>
<p> </p>
<br clear="all" />
<p>第一步：请填写下面的充值定单。 <br />
  第二步：网上银行支付。<br />
  当您提交了定单后，将被提示进入相应的网站进行安全支付。<br />
  如果您的银行卡还没有开通此类支付服务，请按照说明和提示，开通支付业务。<br />
  第三步：如果您已经完成了支付的流程，则充值即可成功。<br />
</p>
<div>
  <p>链接到个人管理的“充值记录”页面</p>
</div>
当您完成了第二步的支付后，相应的金币会自动存入您的账号中。<br />
您可以随时查看所下定单的充值的情况
</p>
               </div>
            </div>
```

```
        </div>
    </div>
</div>
</body>
</html>
```

【代码解读】

操作提示区的作用是在用户充值的过程中显示提示信息，帮助用户完成充值流程。合理的操作提示引导可以帮助用户更好地完成账户充值，提升用户体验。

12.4.2　充值引导程序

充值引导程序的需要实现的功能有两个：

（1）获得当前用户的账户余额；

（2）加载充值引导页面。

为了便于用户直观地了解提交信息的情况，在文件最后插入了一段 HTML 页面监测代码。创建文件名为 pay.php 的 PHP 脚本程序，具体代码如下：

```
<?php
/*******************************************
'文件名：pay.php
'主要功能：支付引导程序
'说明：
*******************************************/
require once (dirname (   FILE   ) . "/inc/inc.read.dbconfig.php");
                                          //数据库配置文件
require once (dirname (   FILE   ) . "/include/include.functions.php");
                                          //公共函数库文件
include once (dirname (   FILE   ) . "/include.smarty.php");
                                          //加载模板引擎配置文件
$mysql  =  mysql connect  (  $mysqlconf  ["host"],  $mysqlconf  ["user"],
$mysqlconf ["pass"] ) or die ( 'Not connected : ' . mysql error () );
mysql_select_db ( $mysqlconf ["db"], $mysql ) or die ( 'Can\'t use foo :
' . mysql error () );
$uids = $ GET ['uid'];
$Surplus = Surplus ( $uids, $mysql );     //当前用户账户剩余货币数量
//确认规则!
$orderid = time ();                       //生成统一的订单编号
$smarty->assign ( "uid", $uids );
$smarty->assign ( "orderid", $orderid );  //生成订单变量的 Smarty 标签
$smarty->assign ( "Surplus", $Surplus );  //生成剩余货币变量的 Smarty 标签
$smarty->display ( "./pay.html" );        //显示输出模板
mysql close ( $mysql );                   //关闭数据库连接
?>
```

【代码解读】

以上代码的作用是支付引导。这里需要注意的是，uid 需要通过 Get 的方式获得，读者可以根据具体的需要，创建自动获得当前 uid 的函数。

在浏览器中执行下面的链接：

```
http://localhost/payment/pay.php?uid=10
```

在浏览器中预览如图 12.2 所示。

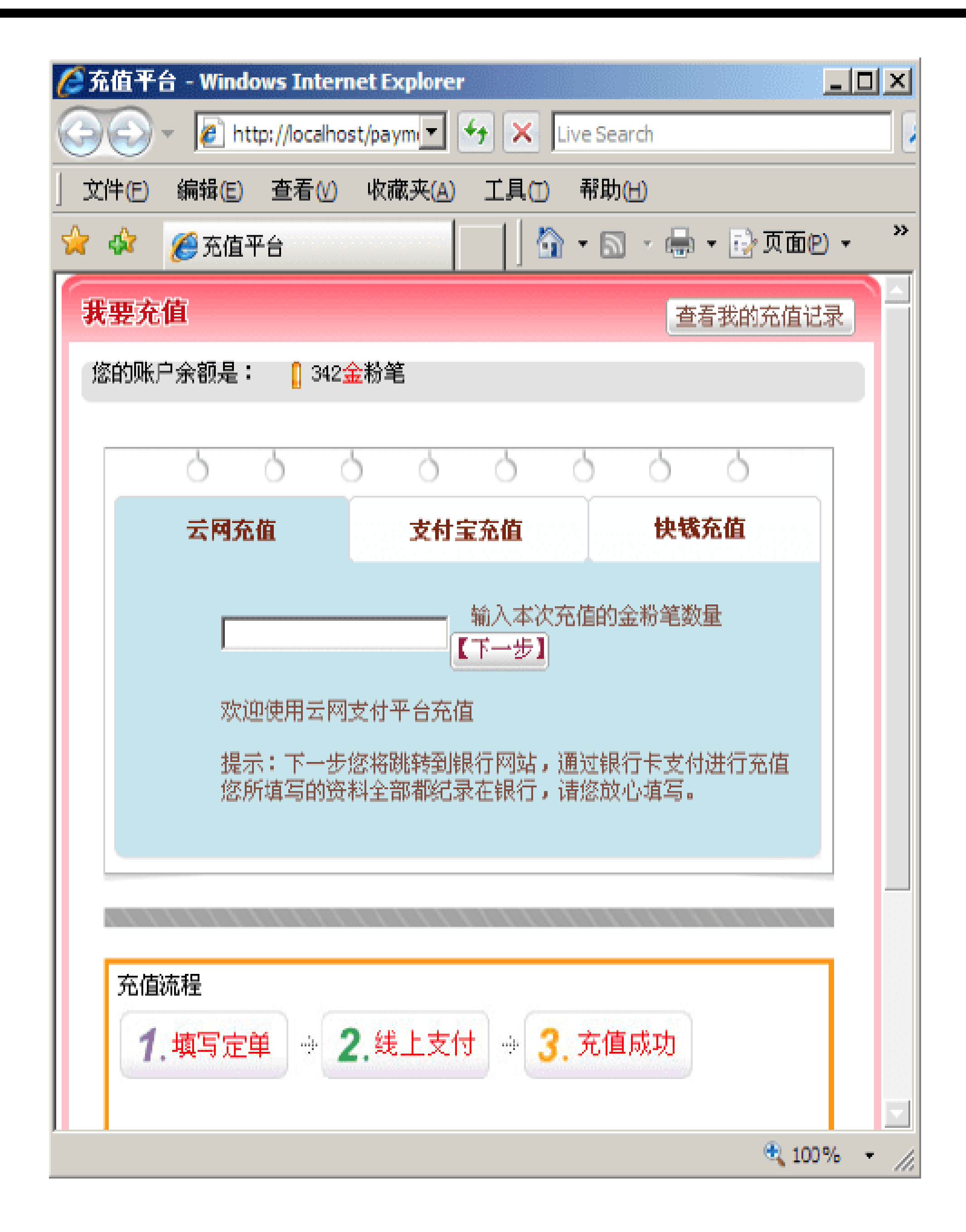


图 12.2　充值引导预览

12.5　服务端充值接口

用户通过充值引导页面输入充值金额后，服务端充值接口程序会分 3 个步骤完成充值过程：

（1）将表单信息提交给订单提交程序，处理成支付网关需要的表单格式，并发送给支付网关验证。

（2）验证通过后，支付网关会调用程序预先设定通知地址，程序会将通知反馈结果以 XML 的形式输出在本页，支付网关会解析输出结果。

（3）完成接收支付网关支付通知信息，验证信息有效性，同时为用户显示支付结果和相关订单信息，更新数据库完成充值。

12.5.1　订单确认

订单确认程序完成将订单信息提交至支付网关的功能，按支付网关要求的格式将订单信息提交至支付接口，进行支付操作。创建文件名为 SendOrder.php 的 PHP 脚本文件，负责完成将订单信息提交至支付网关的功能，代码内容如下：

```
<?
/*******************************************
*文件名：SendOrder.php
*主要功能：完成将订单信息提交至支付网关的功能
*描述：支付网关要求的格式将订单信息提交至支付接口，进行支付操作
'*******************************************/
require_once("../inc/inc.read.dbconfig.php");      //数据库配置文件
```

```
require_once("../include/include.functions.php");  //公共函数库文件
$mysql    =    mysql connect($mysqlconf["host"],    $mysqlconf["user"],
$mysqlconf["pass"])or die ('Not connected : ' . mysql error());
mysql select db($mysqlconf["db"], $mysql)or die ('Can\'t use foo : ' .
mysql error()
$uids=var_process('get','uid');                     //获得当前用户的 uid
$c_order=var_process('post','orderid');             //获得充值的必要参数
$c orderamount= var process('post','textfield');    //充值金额
$record['uid']=$uids;//用户 id
$record['transtype']='1';                           //交易类型 0 消费，1 充值
$record['score']=$c_orderamount;                    //金额
$record['desctext']=$desctext;                      //交易描述
//如果记录交易数据成功会返回 true，失败返回 false
$Transaction_record=Transaction_record($record,$mysql);
//对 account detail 操作，记录当前交易记录
if($Transaction_record=='true'){
    $cid=mysql insert id();
    //记录订单号和对应的平台
    Charge order($cid,1,$c order,$mysql);
    //给账户记录交易次数
    //ount user num($uids,$mysql);
    }else{
    echo "-1\n";//报错
    exit;
}
$merchant param=$uids;
$merchant_param.=".";
$merchant param.=$cid;
//--订单信息---
    $c mid      = "1033562";                //编号，在申请成功后即可获得，可以
                                              在申请成功的邮件中获取该编号
    $c name     = "张三";                   //订单中的收货人姓名
    $c_address  = "北京市朝阳区 XX";        //订单中的收货人地址
    $c tel      = "010-12345678";           //订单中的收货人电话
    $c_post     = "100001";                 //订单中的收货人邮编
    $c_email    = "YourEmail@HostName.com";//订单中的收货人 Email
    //$c orderamount = "0.01";              //订单总金额
    $c_ymd      = "20100530";   //订单的产生日期,格式为"yyyymmdd",如 20050102
    $c moneytype= "0";                      //支付币种，0 为人民币
    $c_retflag  = "1";//订单支付成功后是否需要返回指定的文件，0：不用返回 1：需要返回
    $c_paygate  = "";
    $c returl   = "./PortSample/GetPayNotify.php";//如果 c retflag 为 1 时，
    该地址代表接收支付结果通知的页面，请提交完整文件名(对应范例文件：GetPayNotify.php)
    $c memo1=$merchant param;               //需要在支付结果通知中转发的参数一
    $c_memo2    = "12345";                  //需要在支付结果通知中转发的参数二
    //正确的支付密钥
    $c pass     = "f3zmsjk9r7"; //支付密钥，请登录管理后台，在账户信息-基本信息-
                                  安全信息中的支付密钥项
    $notifytype = "1";  //0 普通通知方式，1 服务器通知方式，空值为普通通知方式
    $c_language = "0";  //对启用了国际卡支付时，可使用该值定义消费者在银行支付时的
                          页面语种，值为：0 银行页面显示为中文，1 银行页面显示为英文
    $srcStr = $c mid . $c order . $c orderamount . $c ymd . $c transnum .
    $c moneytype . $c memo1 . $c memo2 . $c pass;
    //--对订单信息进行 MD5 加密
    //对订单信息进行 MD5 签名后的字符串
?>
```

【代码解读】

以上的代码负责完成将订单信息提交至支付网关的功能，由于支付网关提交参数是固定的，因此需要做的就是将对应的参数信息提取出来，并为对应的变量（订单参数）赋值。

对应的页面模板代码和数据显示标签（Smarty 数据标签）如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>充值确认</title>
<link href="./style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div class="layer">
<h3><a href="#"><img src="../img/laydot.gif" alt=""onclick="window.clock()"
/></a>充值中心</h3>
<h4>充值确认</h4>
<table width="100%" border="0">
    <tr>
        <td width="39%" height="22" align="right">订单号码：</td>
        <td width="61%" class="hui"><?=$c_order?></td>
    </tr>
    <tr>
        <td height="22" align="right">充值用户：</td>
        <td class="red"><?=$c_name?></td>
    </tr>
    <tr>
        <td height="22" align="right">充值方式：</td>
        <td class="hui">银行卡充值</td>
    </tr>
    <tr>
        <td height="22" align="right">充值金额：</td>
        <td class="red"><?=$c_orderamount?>元</td>
    </tr>
    <tr>
        <td colspan=2>
        <table width="100%" border="0" align="center" cellpadding="0"
            cellspacing="0">
            <tr>
                <td align="center">
                <form name="payForm1"action="https://www.cncard.net/
                purchase/getorder.asp" method="POST">
                <input type="hidden" name="c_mid" value="<?=$c_mid?>">
                <input type="hidden" name="c_order" value="<?=$c_order?>">
                <input type="hidden" name="c_name" value="<?=$c_name?>">
                <input type="hidden" name="c_address" value="<?=$c_
                address?>">
                <input type="hidden" name="c_tel" value="<?=$c_tel?>">
                <input type="hidden" name="c_post" value="<?=$c_post?>">
                <input type="hidden" name="c_email" value="<?=$c_email?>">
                <input type="hidden" name="c_orderamount" value="<?=$c_
                orderamount?>">
                <input type="hidden" name="c_ymd" value="<?=$c_ymd?>">
                <input type="hidden" name="c_moneytype" value="<?=$c_
                moneytype?>">
                <input type="hidden" name="c_retflag" value="<?=$c_retflag?>">
                <input type="hidden" name="c_paygate" value="<?=$c_paygate?>">
                <input type="hidden" name="c_returl" value="<?=$c_returl?>">
                <input type="hidden" name="c_memo1" value="<?=$c_memo1?>">
```

```
                <input type="hidden" name="c memo2" value="<?=$c memo2?>">
                <input type="hidden" name="c language" value="<?=$c
                language?>">
                <input type="hidden" name="notifytype" value="<?=
                $notifytype?>">
                <input type="hidden" name="c signstr" value="<?=$c signstr?>">
        <div class="hint"><span>提示：</span>请确认以上信息正确后点击确认按钮
        </div>
                <div class="button"><a href="#" onclick="submit()"> <img
                    src="../img/laybutton01.gif" alt="" /> </a></div>
                </form>
                </td>
            </tr>
        </table>
        </td>
    </tr>
</table>
</div>
</body>
</html>
```

在支付引导页面填写充值金额后，单击“下一步”按钮进入订单确认页面，效果如图 12.3 所示。

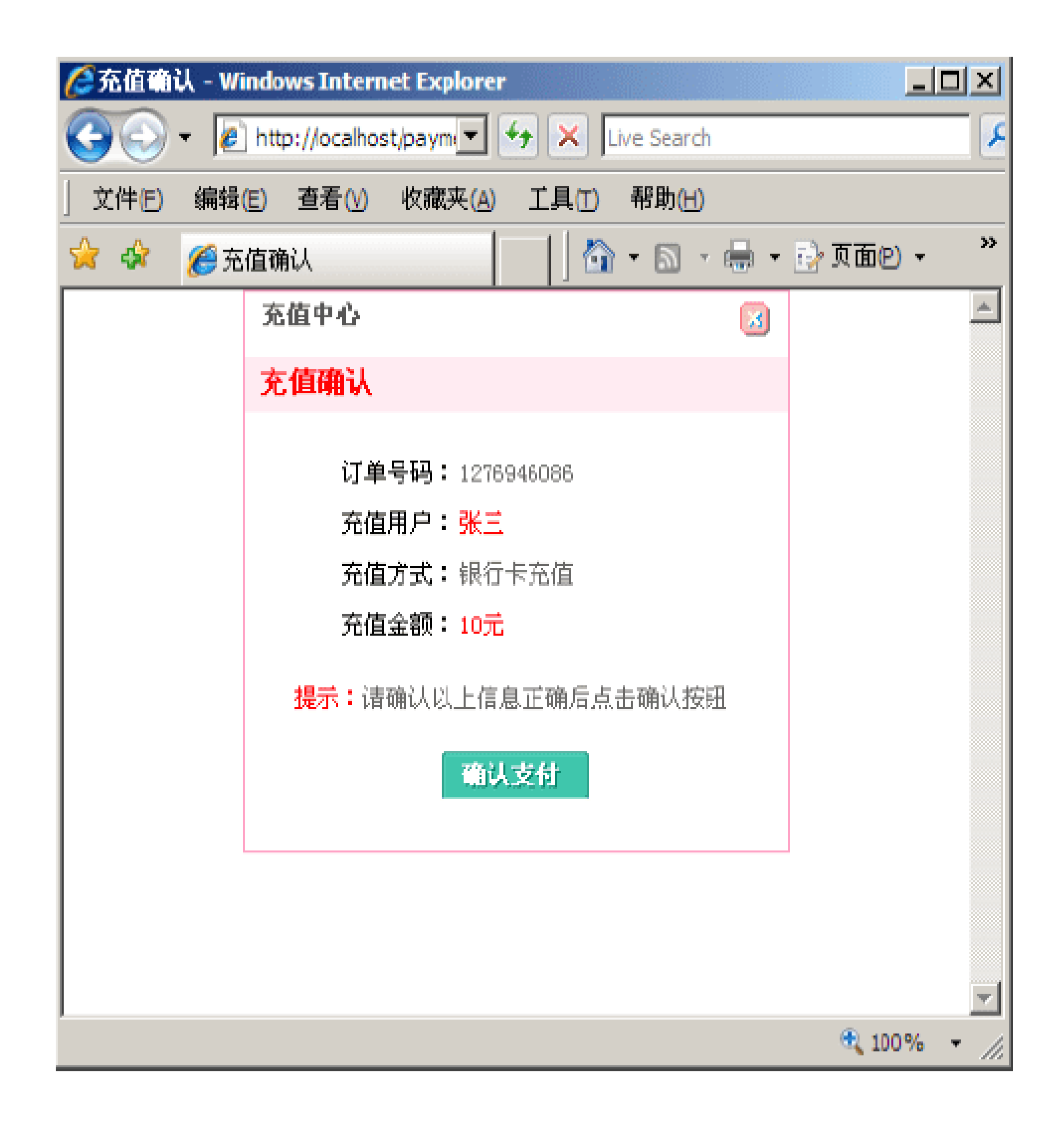


图 12.3　订单确认页面

在订单确认页面单击“确认支付”按钮进入第三方公司的选择支付页面，在这里可以选择支付的类型（网上银行支付）并选择具体的网上银行。然后跳转到该银行的网上支付页面，进行网上支付。

12.5.2　接收网关支付通知

当网上银行提示“扣费成功”后，会向第三方的支付网关返回已支付的通知，这时支付网关会调用客户预先设定好的返回验证程序，来接收和验证支付信息是否正确。当验证结束后，程序会直接将通知反馈结果以 XML 的形式输出在该页，支付网关会解析输出结果，输出的格式为：

```
<result>1</result><reURL> http://localhost/payment/GetPayHandle.php</reURL>
```

创建文件名为 GetPayNotify.php 的 PHP 脚本文件完成上述功能，文件中的代码如下：

```
<?
/********************************************
 *文件名：GetPayNotify.php
 *主要功能：完成接收支付网关支付通知信息，验证信息有效性，判断支付结果功能
 *说明：
 * 1.本页面请不要使用诸如 response.redirect 等页面转向的语句
 * 2.请直接将通知反馈结果以 XML 的形式输出在本页，支付网关会解析输出结果
 ********************************************/
//--获取支付网关向商户发送的支付通知信息(以下简称为通知信息)
$c_mid = $_REQUEST ['c_mid']; //商户编号，在申请商户成功后即可获得，可以在申请商
                                户成功的邮件中获取该编号
$c_order = $_REQUEST ['c_order'];          //商户提供的订单号
$c_orderamount = $_REQUEST ['c_orderamount'];
                   //商户提供的订单总金额，以元为单位，小数点后保留两位，如 13.05
$c_ymd = $_REQUEST ['c_ymd'];  //商户传输过来的订单产生日期，格式为"yyyymmdd",
                                如 20050102
$c_transnum = $_REQUEST ['c_transnum'];    //支付网关提供的该笔订单的交易流水
                                            号，供日后查询、核对使用；
$c_succmark = $_REQUEST ['c_succmark'];    //交易成功标志，Y-成功 N-失败
$c_moneytype = $_REQUEST ['c_moneytype'];  //支付币种，0 为人民币
$c_cause = $_REQUEST ['c_cause'];          //如果订单支付失败则该值代表失败原因
$c_memo1 = $_REQUEST ['c_memo1']; //商户提供的需要在支付结果通知中转发的商户参数一
$c_memo2 = $_REQUEST ['c_memo2']; //商户提供的需要在支付结果通知中转发的商户参数二
$c_signstr = $_REQUEST ['c_signstr']; //支付网关对以上信息进行 MD5 加密后的字符串
//--将获得的通知信息拼成字符串，作为准备进行 MD5 加密的源串，需要注意的是，在拼串时，
  先后顺序不能改变
//商户的支付密钥，登录商户管理后台(https://www.cncard.net/admin/)，在管理首页可
  找到该值
$c_pass = "f3zmsjk9r7";
$srcStr = $c_mid . $c_order . $c_orderamount . $c_ymd . $c_transnum .
$c_moneytype . $c_memo1 . $c_memo2 . $c_pass;
//--对支付通知信息进行 MD5 加密
$r_signstr = md5 ( $srcStr );
//--校验商户网站对通知信息的 MD5 加密的结果和支付网关提供的 MD5 加密结果是否一致
if ($r_signstr != $c_signstr) {
    echo "签名验证失败";
    exit ();
}
//--校验商户编号
$MerchantID = "1033562"; //商户自己的编号
if ($MerchantID != $c_mid) {
    echo "提交的商户编号有误";
    exit ();
}
//--根据返回的支付结果，商户进行自己的发货等操作
if ($c_succmark = "Y") {
    //根据商户自己商务规则，进行发货等系列操作
    //echo "当确认支付以后，进行发货等系列操作...";
    //--输出支付结果通知反馈
    //<result>：值固定为 1，表示商户已成功收到网关的支付成功的通知
    //<reURL> : 商户显示给用户处理结果页面的 URL(对应范例文件：GetPayHandle.php)
```

```
    echo "<result>1</result><reURL>./PortSample/GetPayHandle.php</reURL>";
}
?>
```

注意：由于没有真的使用网银充值，所以无法通过网关调用该程序。但作为调试，可以在浏览器中输入下面的地址模拟支付网关的调用过程，来达到调试调用程序的目的。

```
http://localhost/payment/PortSample/GetPayNotify.php?c_mid=1033562&c_order=20100619&c_orderamount=18&c_ymd=20100530&c_transnum=&c_succmark=N&c_moneytype=0&c_signstr=6f1ac09398f03c48749a0e38d872fcd8&c_memo1=10.32&c_memo2=12345
```

在浏览器中执行输出：

```
<result>1</result><reURL>./PortSample/GetPayHandle.php</reURL>
```

从以上的输出中可以得知调试通过，支付成功。

12.5.3　支付结果显示

当支付网关解析 XML 格式的返回值后，会通过服务器间通知的方式，调用 XML 中的程序地址。该程序会完成接收支付网关支付通知信息，验证信息有效性为用户显示支付结果（更新数据库中的账户金额完成平台的支付）和相关订单信息等。

创建文件名为 GetPayHandle.php 的 PHP 脚本文件，用来完成上述功能，代码如下：

```
<?
/********************************************
 *文件名：GetPayHandle.php
 *主要功能：完成接收支付网关支付通知信息，验证信息有效性，给用户显示支付结果和相关订单信息等
 *说明：
 * 1.本页面请不要使用诸如 response.redirect 等页面转向的语句
 * 2.可根据支付结果和订单号查询出本次订单信息或发货信息显示给用户
 * 3.本页面如果含有图片、样式或链接，请将路径或地址包括域名，比如<img src="http://www.yoursitename.com/imagers/aaa.gif">
 *调试链接：http://localhost/payment/PortSample/GetPayHandle.php?c_mid=1033562&c_order=20100619&c_orderamount=18&c_ymd=20100530&c_transnum=&c_succmark=Y&c_moneytype=0&c_signstr=6f1ac09398f03c48749a0e38d872fcd8&c_memo1=10.32&c_memo2=12345
 ********************************************/
require_once ("../inc/inc.read.dbconfig.php");    //数据库配置文件
require_once ("../include/include.functions.php");//公共函数库文件
//--连接数据库
$mysql = mysql_connect ( $mysqlconf ["host"], $mysqlconf ["user"], $mysqlconf ["pass"] ) or die ( 'Not connected : ' . mysql_error () );
mysql_select_db ( $mysqlconf ["db"], $mysql ) or die ( 'Can\'t use foo : ' . mysql_error () );
//--获取支付网关向发送的支付通知信息(以下简称为通知信息)
$c_mid = $_REQUEST ['c_mid'];
                    //编号，在申请成功后即可获得，可以在申请成功的邮件中获取该编号
$c_order = $_REQUEST ['c_order'];              //提供的订单号
$c_orderamount = $_REQUEST ['c_orderamount'];  //提供的订单总金额，以元为单位，
                                                 小数点后保留两位，如 13.05
$c_ymd = $_REQUEST ['c_ymd'];
```

```
                        //传输过来的订单产生日期，格式为"yyyymmdd"，如 20050102
$c transnum = $ REQUEST ['c transnum'];
                        //支付网关提供的该笔订单的交易流水号，供日后查询、核对使用；
$c succmark = $ REQUEST ['c succmark'];    //交易成功标志，Y-成功 N-失败
$c_moneytype = $_REQUEST ['c_moneytype'];  //支付币种，0 为人民币
$c_cause = $_REQUEST ['c_cause'];       //如果订单支付失败，则该值代表失败原因
$c memo1 = $ REQUEST ['c memo1'];       //提供的需要在支付结果通知中转发的参数一
$c_memo2 = $_REQUEST ['c_memo2'];       //提供的需要在支付结果通知中转发的参数二
$c signstr = $ REQUEST ['c signstr'];  //支付网关对已上信息进行 MD5 加密后的字符串
//--将获得的通知信息拼成字符串，作为准备进行 MD5 加密的源串，需要注意的是，在拼串时，
  先后顺序不能改变的支付密钥，在管理首页可找到该值
$c pass = "f3zmsjk9r7";
$srcStr = $c mid . $c order . $c orderamount . $c ymd . $c transnum .
$c_moneytype . $c_memo1 . $c_memo2 . $c_pass;
//--对支付通知信息进行 MD5 加密
$r signstr = md5 ( $srcStr );
//--校验网站对通知信息的 MD5 加密的结果和支付网关提供的 MD5 加密结果是否一致
if ($r_signstr != $c_signstr) {
    echo "签名验证失败";
    exit ();
}
//--校验编号
$MerchantID = "1033562"; //商户的编号
if ($MerchantID != $c_mid) {
    echo "提交的编号有误";
    exit ();
}
if ($c succmark = "Y") {
    //程序内部调用扣费函数
    //由于在 GetPayNotify.php 中已进行了实际发货操作,所以在此处只要系统中查询出本订
      单的结果，然后给用户显示购买成功的提示信息即可
    $score = $c orderamount;
    $pieces = explode ( ".", $c memo1 );
    $uid = $pieces [0]; //uid
    $cid = $pieces [1]; //cid
    //防止刷成功页面给对应账号充值
    //Update_status_record 会将对应的 status 设置成 1,只有当 status=0(未充值状态)
      的时候才会返回 true
    //确认用户付费成功了之后将 account_detail 中的 status 设置为 1（表示充值成功）0
      为失败
    $Update status record = Update status record ( $cid, $mysql );
    //增减财富的操作
    if ($Update status record == 'true') {
        //对 user account 表进行操作，增加货币数量
        $Charge_num = Charge_num ( $uid, $score, $mysql );
        if (! empty ( $Charge num )) {
            //充值金额
            $score t = $score;
        } else {
            return false;
        }
        //--释放记录集，关闭数据库连接
        //mysql free result($result);
        mysql close ( $mysql );
    } else {
```

```
        echo "-1\n"; //表示用户刷新页面
    }
}
?>
```

【代码解读】

上述代码的核心功能是完成接收支付网关支付通知信息，验证信息有效性，给用户显示支付结果和相关订单信息等。在确认用户付费成功之后，调用公共函数库$Update_status_record=Update_status_record()函数将数据库库中的 account_detail 表中的 status 设置为 1（表示充值成功），这样就可以防止充值页面被反复刷新。然后在通过$Charge_num= Charge_num()函数对 user_account 表进行操作，完成账户充值的功能。

对应的页面显示模板代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>支付结果显示</title>
<meta name="keywords" content="支付结果显示" />
<meta name="description" content="支付结果显示" />
<link href="../css/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="Housecenter">
<div id="shop">
    <div class="Waretop">
        <span><img src="../img/wptopr.gif" /></span>
              <img src="../img/wptopl.gif" />
    </div>
<div class="Waretopd">
<div class="Waretopdw">充值成功</div>
</div>
<div class="Warecnt shopLeft">
<div class="LeftAsse">
    <div class="LeftAssetop">
        <span><img src="../img/glR.gif" /></span>
              <img src="../img/glL.gif" width="34" height="24" />
    </div>
<div class="LeftAssecnt">
<div class="LeftAsseText">
    <div class="LeftAsseTextT" style=" margin-top:18px; text-align:center;
    color:#ff0000;">
        <?php
            if(!empty($score t)){
                echo "恭喜 您的充值已成功! 充值金额".$score_t;
                }else{
                echo "充值异常: (";
            }

        ?>
    </div>
</div> </div> </div> </div> </div> </div>
</body>
</html>
```

这里，同样可以通过链接的方式，模拟服务期间通知的模式来验证程序，模拟用的链接地址如下：

```
http://localhost/payment/PortSample/GetPayHandle.php?c_mid=1033562&c_or
der=20100619&c_orderamount=18&c_ymd=20100530&c_transnum=&c_succmark=Y&c
_moneytype=0&c_signstr=6f1ac09398f03c48749a0e38d872fcd8&c_memo1=10.32&c
_memo2=12345
```

在浏览器中执行上面的链接，效果如图 12.4 所示。

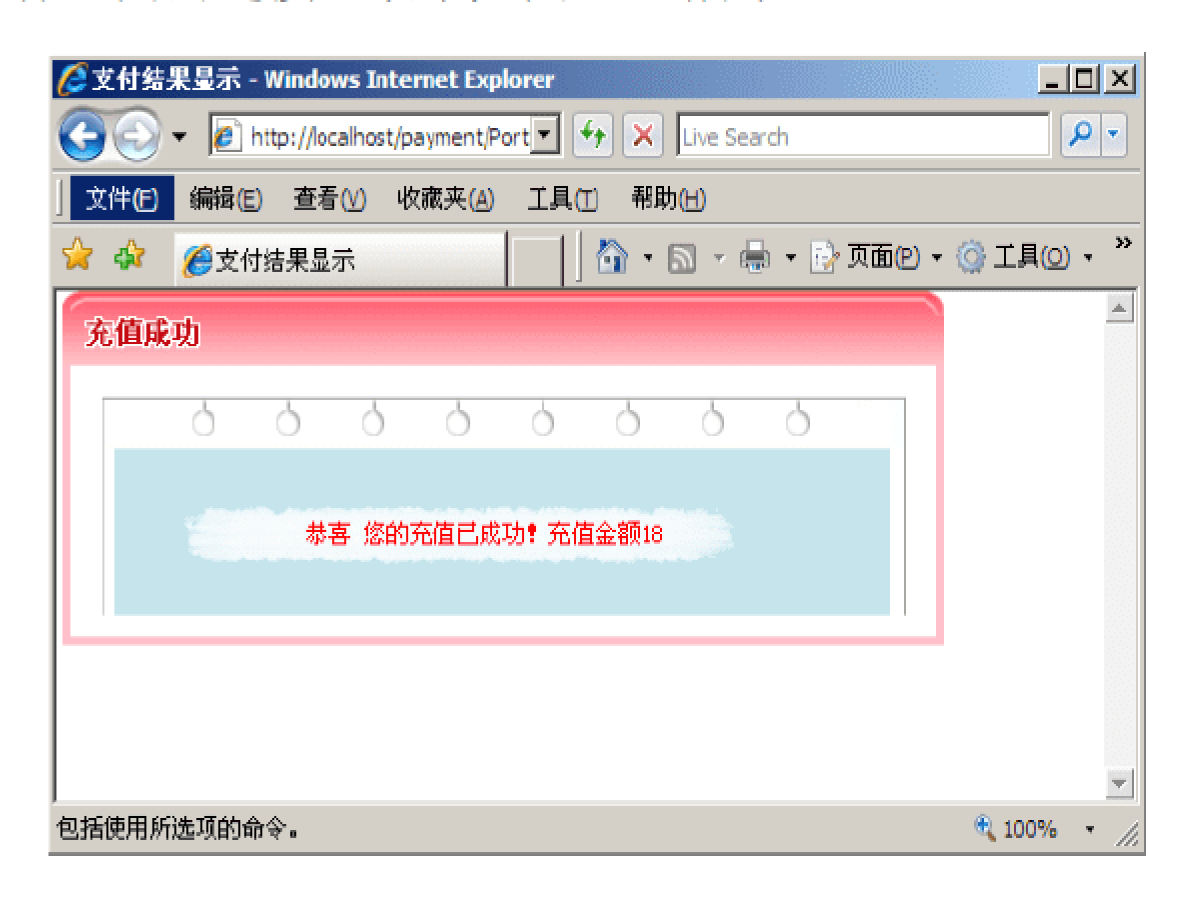


图 12.4　充值结果页面

12.6　小　　结

本章从充值平台的实现目的讲起，介绍了使用第三方支付接口进行开发调试的流程，第 12.2 节对第三方支付接口的使用做了具体的说明，为下面的程序搭建和调试做了必要的准备。在第 12.3 节介绍了通知支付结果方式的实现，其目的是直观地梳理了本地程序和支付网关的互动的流程。其中的服务器间通知方式是本节的重点，它是这类针对支付网关编程的核心。第 12.4 节创建了充值引导页面用来填写基本的充值信息，同时也是充值平台模块的入库文件。第 12.5 节是充值平台模块的核心内容。通过 3 个步骤与支付网关的交互，实现了安全的支付最终完成充值。

本章的目的是使读者了解如何通过对第三方支付网关的编程，实现对网站用户的充值。该过程是典型的电子交易过程，在程序方面还复习了 Smarty 模板引擎的使用。

第 13 章　在线调查模块（PHP+MySQL+XML）

在线调查是指通过互联网及其调查系统把传统的调查和分析方法在线化、智能化。典型的在线调查系统包括 3 个部分，即客户、调查系统、参与人群。在线调查模块是一个实用性和互动性很强的程序模块，通过该模块读者可以为网站定制调查内容，并查询相应的调查结果。本章主要介绍在线调查模块的实现原理和程序的逻辑组成。通过本章的学习，读者可以复习 PHP 脚本与 MySQL 数据库的交互的相关内容、学习使用 XML 格式文档作为数据存储的中间载体实现数据存储，以及学习 HTML 页面标签实现数据的直观显示（如柱形图）等。

本章主要涉及的知识点如下。

- mysql_query("SET NAMES utf8");：指定程序从数据库中读取数据时的字符集，避免乱码的出现。
- XML 格式文档：标准通用标记语言，是依赖于内容的技术，是当前处理结构化文档信息的有力工具。
- simplexml_load_string();：是 PHP 中处理 XML 格式文档的常用函数。
- header('Content-Type: text/xml');：使用 PHP 生成 XML 格式文档需要使用 header() 函数向浏览器声明文件的类型，防止出错。

13.1　在线调查系统需求分析

在线调查系统是一个利用网络进行信息调查反馈的系统，它取代了传统的手工式的信息调查（如写信、打电话等），方便网站及公司业务部门信息收集和反馈，提高了反馈效率，为改进用户体验提供了需求基础。在线调查系统可以分成前端展示和后台管理两个部分。

（1）前端展示部分包括调查页面显示（view.php）、调查数据保存（vote.php）和调查结果查看（result.php）。

（2）后台管理部分包括调查管理界面显示（admin.php）和修改调查项目（update.php）。

通过图 13.1 可以直观地了解在线调查系统的组成和各个“部件”之间的关系。

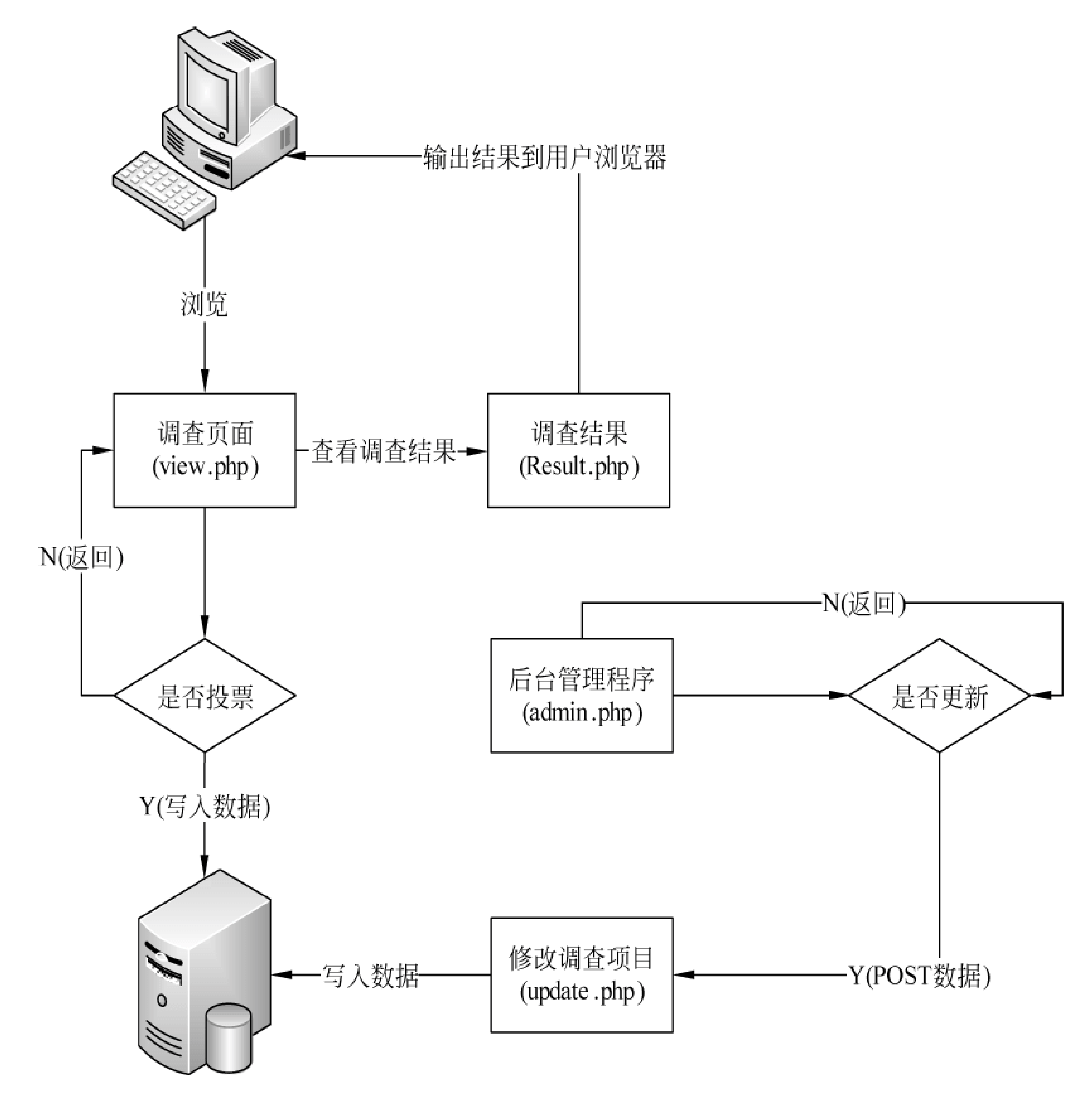


图 13.1　在线调查系统程序流程

13.2　数据库设计

在线调查模块数据库的设计目的是存储调查相关的信息，每一条记录代表一个调查专题，增加新的专题就新增一条记录。改变调查内容时，更新该条记录对应的字段即可。这里调查内容部分涉及的信息量较多采用 XML 格式作为标准的交互数据格式。

在线调查部分的数据库中，只用到了一张调查存储表（vote）。该表是用来存储调查基本信息和调查内容的，提供后台管理使用和前台页面展示。了解功能划分之后，下面来看具体的设计步骤：

（1）建立 cms_vote 数据库，SQL 语句如下：

```
CREATE DATABASE 'cms_vote' ;
```

（2）在 cms_vote 数据库中，建立一个命名为 vote 的调查存储表，结构如表 13.1 所示。创建表及相关字段的 SQL 语句如下：

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
-- 数据库: 'cms_vote'
--

-- --------------------------------------------------------
```

```
--
-- 表的结构 'vote'
--

CREATE TABLE 'vote' (
  'aid' int(11) NOT NULL auto increment,
  'votename' varchar(50) NOT NULL,
  'starttime' date NOT NULL,
  'endtime' date NOT NULL,
  'totalcount' int(11) NOT NULL default '0',
  'votenote' text NOT NULL,
  PRIMARY KEY ('aid')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;
```

表 13.1　调查存储表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
aid	int	11	Not null	auto_increment	调查编号（自增主键）
Votename	varchar	50	Not null		调查主题
Starttime	Date		Not null		调查开始时间
Endtime	Date		Not null		调查结束时间
Totalcount	int	11	Not null	0	调查总数
Votenote	Text		Not null		调查内容（XML）

注意：以上调查存储表可以满足基本的在线调查需求，要想搭建业务逻辑更为复杂的应用，可以通过增加附表的方式实现（例如权限、角色等）。

（3）为了便于后面程序的调试，向数据库中插入一条测试数据，插入测试代码的 SQL 语句如下：

```
--
-- 导出表中的数据 'vote'
--

INSERT INTO 'vote' VALUES (1, '您预计本届世界杯谁会夺冠？', '0000-00-00',
'2010-06-21', 717, 0, '<movies>\r\n<movie>\r\n <title>您预计本届世界杯谁会夺
冠？</title>\r\n<id>1</id>\r\n
    <count>350</count>\r\n<content>巴西</content>\r\n</movie>\r\n<movie>
    \r\n<title>您预计本届世界杯谁会夺冠？</title>\r\n  <id>2</id>\r\n
    <count>47</count>\r\n  <content>西班牙</content>\r\n </movie>\r\n
    <movie>\r\n <title>您预计本届世界杯谁会夺冠？</title>\r\n
    <id>3</id>\r\n  <count>118</count>\r\n <content> 德 国 </content>\r\n
    </movie>\r\n  <movie>\r\n<title>您预计本届世界杯谁会夺冠？</title>\r\n
    <id>4</id>\r\n
    <count>122</count>\r\n  <content>朝鲜</content>\r\n </movie>\r\n <movie>
    \r\n <title>您预计本届世界杯谁会夺冠？</title>\r\n <id>5</id>\r\n
    <count>80</count>\r\n <content>荷兰</content>\r\n </movie>\r\n</movies>');
```

13.3　调查管理后台

调查管理程序是在线调查模块的核心程序。管理员通过它可以调整调查内容、修改基

本的调查信息，也可以通过后台查看调查的一些基本数据情况，根据这些情况再做合适的调整。

13.3.1　调查管理后台界面

根据前面的需求分析和数据库设计可以了解到，调查管理界面主要实现两个功能，即显示基本调查信息和编辑调查内容。

下面来看下具体界面的设计，创建文件名为 admin.php 的 HTML 页面文件，文件内容如下：

```
<html>
<head>
<meta http-equiv='Content-Type' content='text/html; charset=utf-8'>
<title>调查管理</title>
<link href='./base.css' rel='stylesheet' type='text/css'>
</head>
<body background='img/allbg.gif' leftmargin='8' topmargin='8'>
<table width="98%" border="0" align="center" cellpadding="3"
    cellspacing="1" bgcolor="#D1DDAA">
    <tr>
        <td height="28" background="img/tbg.gif"><b>调查管理</b></td>
    </tr>
    <tr>
        <td height="200" bgcolor="#FFFFFF" valign="top">
        <form name="form1" method="post" action="update.php">
        <input type="hidden" name="aid" value="1">
        <table width="100%" border="0" cellspacing="4" cellpadding="4">
            <tr>
                <td width="15%" align="center">调查名称：</td>
                <td width="85%">
                    <input name="votename" type="text" id="votename" value=
                    "您预计本届世界杯谁会夺冠？">
                </td>
            </tr>
            <tr>
                <td align="center">调查总人数：</td>
                <td>
                    <input name="totalcount" type="text" id="totalcount"
                    value="717">
                </td>
            </tr>
            <tr>
                <td align="center">开始时间：</td>
                <td>
            <input name="starttime" type="text" id="starttime" value=
            "0000-00-00">
                </td>
            </tr>
            <tr>
                <td align="center">结束时间：</td>
                <td>
            <input name="endtime" type="text" id="endtime" value="2010-
            06-21">
                </td>
            </tr>
```

```
            <tr>
                <td align="center">投 票 项：<br />
                (请按相同的形式来增加或修改节点，其中属性：id 不能重复)</td>
    <td><textarea name="votenote" rows="8" id="votenote"style="width: 80%">
    <movies>
<movie>
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>1</id>
    <count>350</count>
    <content>巴西</content>
</movie>
<movie>
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>2</id>
    <count>47</count>
    <content>西班牙</content>
 </movie>
 <movie>
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>3</id>
    <count>118</count>
    <content>德国</content>
 </movie>
 <movie>
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>4</id>
    <count>122</count>
    <content>朝鲜</content>
 </movie>
 <movie>
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>5</id>
    <count>80</count>
    <content>荷兰</content>
 </movie>
</movies></textarea></td>
            </tr>
            <tr>
                <td height="47"> </td>
                <td><input type="submit" name="Submit" value="保存调查数据">
                </td>
            </tr>
            <tr>
                <td colspan="2"> </td>
            </tr>
        </table>
        </form>
        </td>
    </tr>
</table>
</body>
</html>
```

【代码解读】

上述管理模板包含了调查管理程序的最基本选项，如调查名称、调查人数（已投票）、开始/结束时间、投票的选项。这里请读者注意，自定义的调查选项显示单元，代码样式如下：

```
<movie>
    <title>您预计本届世界杯谁会夺冠？</title>
```

```
    <id>1</id>
    <count>350</count>
    <content>巴西</content>
</movie>
```

13.3.2　调查管理程序

调查管理程序的功能是，将调查管理界面表单中的信息从数据库中提取出来，并输出到页面 HTML 标签中完成信息展示，同时对可能发生的异常情况做逻辑上的判断，达到控制流程的目的。

打开 admin.php 文件，插入处理数据的 PHP 脚本，将表单中的信息替换为 PHP 变量。修改后的 admin.php 具体内容如下：

```
<?php
/******************************************
'文件名：admin.php
'主要功能：调查管理程序
'说明：
'******************************************/
require_once(dirname(__FILE__)."/dbconfig.php");          //数据库配置文件
require_once(dirname(__FILE__)."/include.functions.php"); //公共函数库文件
$mysql = mysql_connect($mysqlconf["host"], $mysqlconf["user"], $mysqlconf
["pass"])or die ('Not connected : ' . mysql_error());
mysql_select_db($mysqlconf["db"], $mysql)or die ('Can\'t use foo : ' .
mysql_error());
mysql_query("SET NAMES utf8");                          //设置默认字符集
$aid=$_GET['aid'];                                      //获得栏目 id
if(!empty($aid)){                                       //判断栏目是否存在
$sql="SELECT * FROM 'vote' WHERE 'aid' ='".$aid."'";
$result = mysql_query($sql);
$row=mysql_fetch_array($result);
    }else{
    echo '<script type="text/javascript">alert(\'该调查异常\')</script>';
    exit;
}
?>
```

【代码解读】

上面的程序创建数据库连接，然后获得指定的栏目 id，执行 SQL 语句从数据库中查找对应的投票信息，如果栏目不存在则进入异常报错流程。

调查管理程序对应的页面模板代码如下：

```
<html>
<head>
<meta http-equiv='Content-Type' content='text/html; charset=utf-8'>
<title>调查管理</title>
<link href='./base.css' rel='stylesheet' type='text/css'>
</head>
<body background='img/allbg.gif' leftmargin='8' topmargin='8'>
<table width="98%" border="0" align="center" cellpadding="3" cellspa
cing="1" bgcolor="#D1DDAA">
  <tr>
    <td height="28" background="img/tbg.gif"><b>调查管理</b></td>
</tr>
```

```
<tr>
    <td height="200" bgcolor="#FFFFFF" valign="top">
    <form name="form1" method="post" action="update.php">
    <input type="hidden" name="aid" value="<?php echo $aid?>">
        <table width="100%" border="0" cellspacing="4" cellpadding="4">
          <tr>
            <td width="15%" align="center">调查名称：</td>
            <td width="85%"> <input name="votename" type="text" id="votename"
            value="<?php echo $row['votename']?>">
            </td>
          </tr>
          <tr>
            <td align="center">调查总人数：</td>
            <td><input name="totalcount" type="text" id="totalcount"
            value="<?php echo $row['totalcount']?>"></td>
          </tr>
          <tr>
            <td align="center">开始时间：</td>
            <td><input name="starttime" type="text" id="starttime" value=
            "<?php echo $row['starttime']?>"></td>
          </tr>
          <tr>
            <td align="center">结束时间：</td>
            <td><input name="endtime" type="text" id="endtime" value=
            "<?php echo $row['endtime']?>"></td>
          </tr>
          <tr>
            <td align="center">投 票 项：<br/>
              (请按相同的形式来增加或修改节点，其中属性：id 不能重复) </td>
            <td><textarea name="votenote" rows="8" id="votenote" style=
            "width:80%"><?php echo $row['votenote']?></textarea>
            </td>
          </tr>
          <tr>
            <td height="47"> </td>
            <td><input type="submit" name="Submit" value="保存调查数据"></td>
          </tr>
          <tr>
            <td colspan="2"> </td>
          </tr>
        </table>
      </form>
      </td>
</tr>
</table>
</body>
</html>
```

这里需要注意的是，mysql_query("SET NAMES utf8")；这条语句，很多读者可能都遇见过数据从数据库中取出后显示为乱码的问题。造成这种问题的原因很多，但最主要的就是数据库的字符集与程序编码的字符集不统一。mysql_query("SET NAMES utf8");函数会将数据以 utf-8 字符集编码从数据库中取出，这样可以有效地避免乱码问题。

在浏览器中执行下面的链接：

```
http://localhost/vote/admin.php?aid=1
```

预览效果如图 13.2 所示。

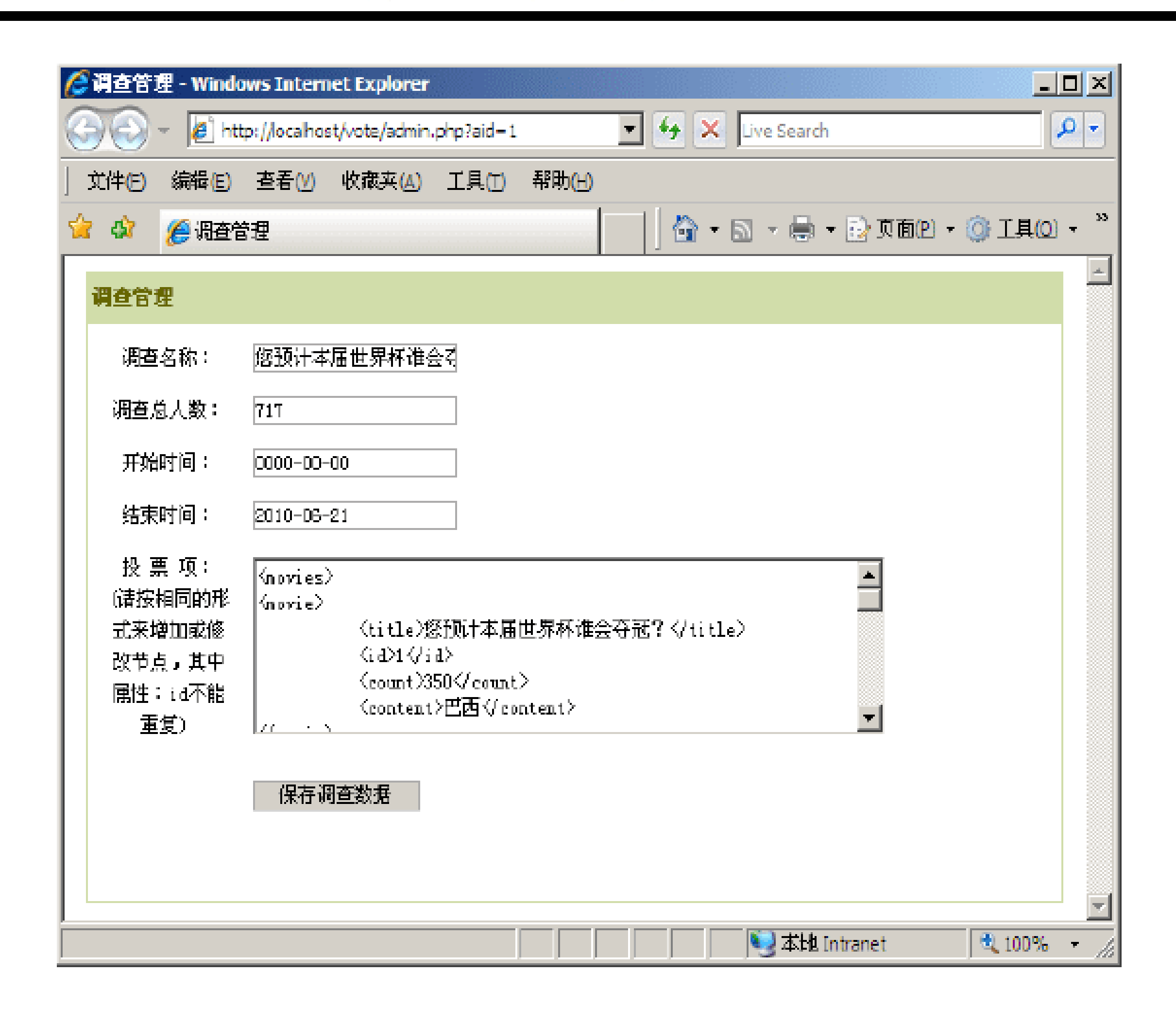


图 13.2　调查管理后台预览

13.3.3　保存调查数据

管理员对调查的基本信息和调查内容做出修改，并单击“保存调查数据”按钮后，会将 admin.php 表单中的信息提交给保存调查数据程序处理。对提交过来的信息做了处理之后，会更新数据库中对应的各个字段的内容，完成本次的修改。

创建文件名为update.php 的 PHP 脚本文件，该脚本文件的功能是处理提交的表单数据，并更新数据库。该文件的具体代码如下：

```
<?php
/*********************************************
'文件名：update.php
'主要功能：保存调查数据程序
'说明：
*********************************************/
require once (dirname (  FILE  ) . "/dbconfig.php");//数据库配置文件
require_once (dirname ( __FILE__ ) . "/include.functions.php");
                                                    //公共函数库文件
$mysql = mysql_connect ( $mysqlconf ["host"], $mysqlconf ["user"],
$mysqlconf ["pass"] ) or die ( 'Not connected : ' . mysql error () );
mysql select db ( $mysqlconf ["db"], $mysql ) or die ( 'Can\'t use foo :
' . mysql_error () );
mysql query ( "SET NAMES utf8" );                   //设置默认字符集
$aid = $_POST ['aid'];                              //获得栏目 id
if (! empty ( $aid )) {                             //判断栏目是否存在
    $sql = "UPDATE 'cms vote'.'vote' SET 'votename' = '" . $ POST
['votename'] . "','starttime' = '" . $ POST ['starttime'] . "','endtime'
= '" . $ POST ['endtime'] . "','totalcount' = '" . $ POST ['totalcount'] .
"','votenote' = '" . $ POST ['votenote'] . "' WHERE 'vote'.'aid' ='" . $aid .
"' LIMIT 1 ;";
    $result = mysql query ( $sql );
    if (! empty ( $result )) {
        echo '<script type="text/javascript">alert(\'数据保存成功\')</script>';
        exit ();
    } else {
        echo '<script type="text/javascript">alert(\'数据保存失败\')</script>';
```

```
        exit ();
    }
} else {
    echo '<script type="text/javascript">alert(\'该调查异常\')</script>';
    exit ();
}
?>
```

在浏览器中执行链接：

```
http://localhost/vote/admin.php?aid=1
```

修改并单击“保存调查数据”按钮，预览效果如图 13.3 所示。

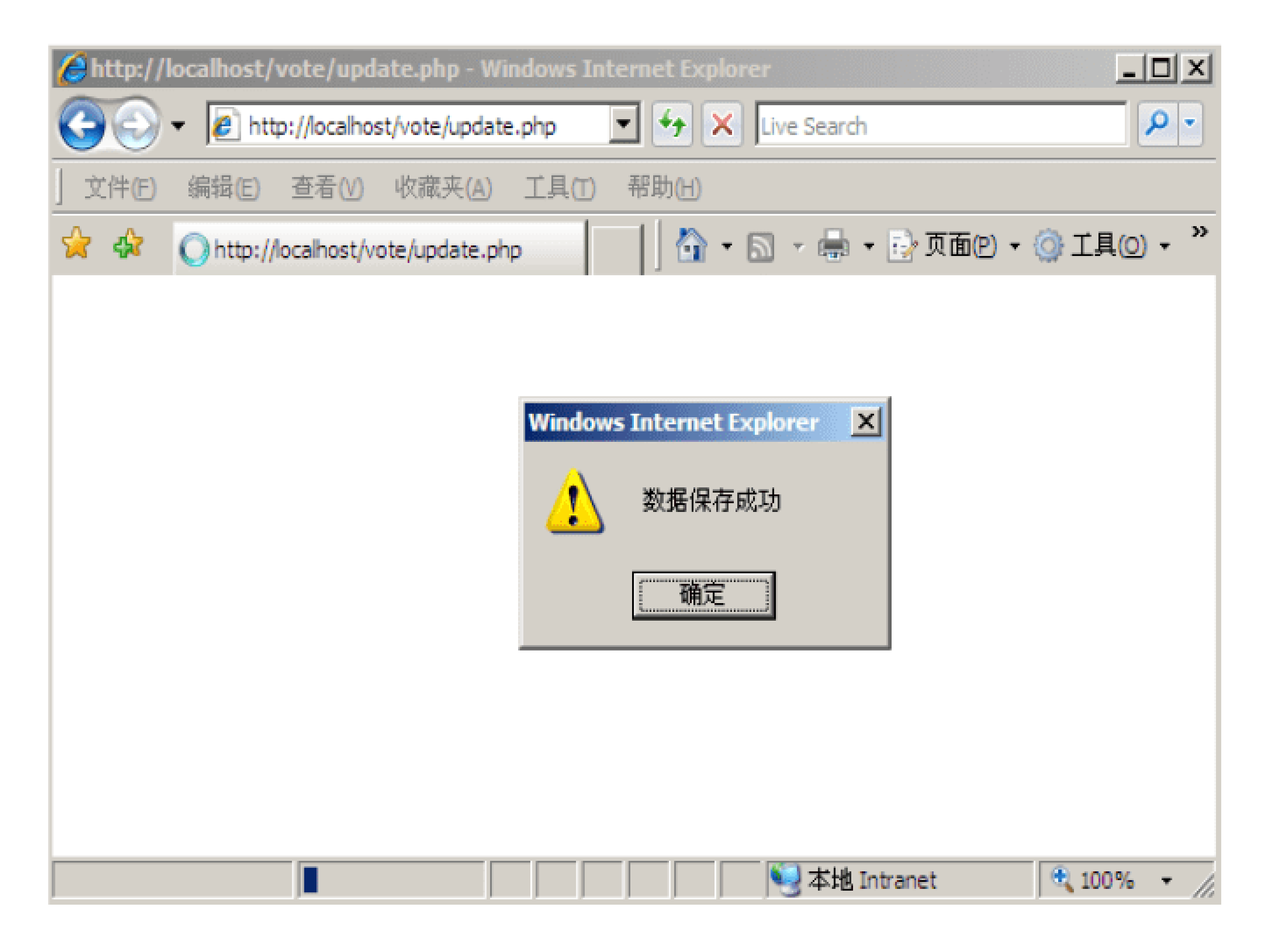


图 13.3　保存调查数据

13.4　在 线 调 查

在线调查的设计目的是，针对当前某一个热点的主题，以提交问卷的方式收集访问者的信息，管理者可以根据调查的结果执行相应的策略，而访问者也通过这种互动的方式表达了自己的观点。

13.4.1　在线调查界面

在线调查的界面的设计要求是，让访问者可以直观地了解被调查的主题，同时最方便和直观地给出自己的观点。这里选择单选按钮这种形式作为调查的主要实现方法。创建文件名为 view.php 的 HTML 页面文件，该文件的代码如下：

```
<HTML xmlns="http://www.w3.org/1999/xhtml">
<HEAD>
<TITLE>在线调查</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="./nspage.css" type=text/css rel=stylesheet>
</HEAD>
```

```
<BODY>
<DIV class="main mceneter">
<DIV class=toplogo>
    <DIV class=title>在线调查 </DIV>
</DIV>
<table width='100%' border='0' cellspacing='1' cellpadding='1' bgcolor=
'#FFFFFF'>
<form name='voteform' method='post' action='vote.php' target=' blank'>
<!--发送调查栏目 id-->
<input type='hidden' name='aid' value='1'>
<tr align='center'>
    <td height='24' bgcolor='#EDEDE2' >您预计本届世界杯谁会夺冠? </td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
        <input type='radio' name='voteitem' value='0'>
        巴西 </td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
    <input type='radio' name='voteitem' value='1'>
    西班牙   </td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
    <input type='radio' name='voteitem' value='2'>
    德国 </td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
    <input type='radio' name='voteitem' value='3'>
    朝鲜 </td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
    <input type='radio' name='voteitem' value='4'>
    荷兰 </td>
</tr>
<tr>
    <td height='24' bgcolor='#FFFFFF'>
    <input type='submit' style='width:40;background-color:#EDEDE2;border:
    1px soild #818279' name='vbt1' value='投票'>
    <input type='button' style='width:80;background-color:#EDEDE2;border:
    1px soild #818279' name='vbt2' value='查看结果' onClick="window.open
    ('result.php?aid=1');">
</td>
</tr>
</form>
</table>
</DIV>
</BODY>
</HTML>
```

【代码解读】

上面的代码需要注意的是表单中单选按钮的设计，请看下面的代码：

```
<input type='radio' name='voteitem' value='0'>
<input type='radio' name='voteitem' value='1'>
<input type='radio' name='voteitem' value='2'>
```

```
<input type='radio' name='voteitem' value='3'>
<input type='radio' name='voteitem' value='4'>
```

这是一个标准的单选按钮标签，这里的设计细节是将所有标签的 name 的值设置成同一个名字（voteitem）。这样在提交给程序处理时，只需要接收 voteitem 的值就可以判断用户选择了哪个选项。

13.4.2　在线调查程序

在线调查程序主要完成的功能有两个。一个是对数据库中的信息做必要的处理，生成变量标签；另一个是输出对应变量值，替换 HTML 页面中调查内容部分。

程序设计部分的难点是对数据中的调查内容字段的处理，在数据库中调查存储表（vote）中的内容数据是以 XML 格式的方式存储的。这样设计的目的是为了通过 XML 格式可以有效地管理信息。因为这部分实际上变成了对调查内容 XML 格式的读写，现在来看 XML 格式的样式和节点说明。

（1）首先定义 4 个节点，分别用标签<title>、<id>、<count>、<content>代表。对应的含义是<title>标签中的内容代表本次调查的主题、<id>标签中的内容代表本次调查的选项序号（id 自增）、<count>标签中的内容代表调查票数、<content>标签中的内容代表调查的内容。

（2）创建一个 xml.php 的 PHP 脚本文档来测试 XML 格式数据。xml.php 中的具体测试代码如下：

```
<?php
/*******************************************
'文件名：xml.php
'主要功能：XML 格式存储调查信息程序
'说明：
'*******************************************/
header('Content-Type: text/xml');
$xmlstr="<movies>
<movie>
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>1</id>
    <count>350</count>
    <content>巴西</content>
</movie>
<movie>
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>2</id>
    <count>47</count>
    <content>西班牙</content>
 </movie>
 <movie>
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>3</id>
    <count>118</count>
    <content>德国</content>
 </movie>
 <movie>
```

```
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>4</id>
    <count>122</count>
    <content>朝鲜</content>
 </movie>
 <movie>
    <title>您预计本届世界杯谁会夺冠？</title>
    <id>5</id>
    <count>80</count>
    <content>荷兰</content>
 </movie>
</movies>";
echo $xmlstr;
?>
```

【代码解读】

以上代码中使用 PHP 脚本生成 XML 文档时，需要在文件头部使用 header()函数来向浏览器声明文件格式。具体的代码如下：

```
header('Content-Type: text/xml');
```

在浏览器中执行链接：

```
http://localhost/vote/xml.php
```

预览效果如图 13.4 所示。

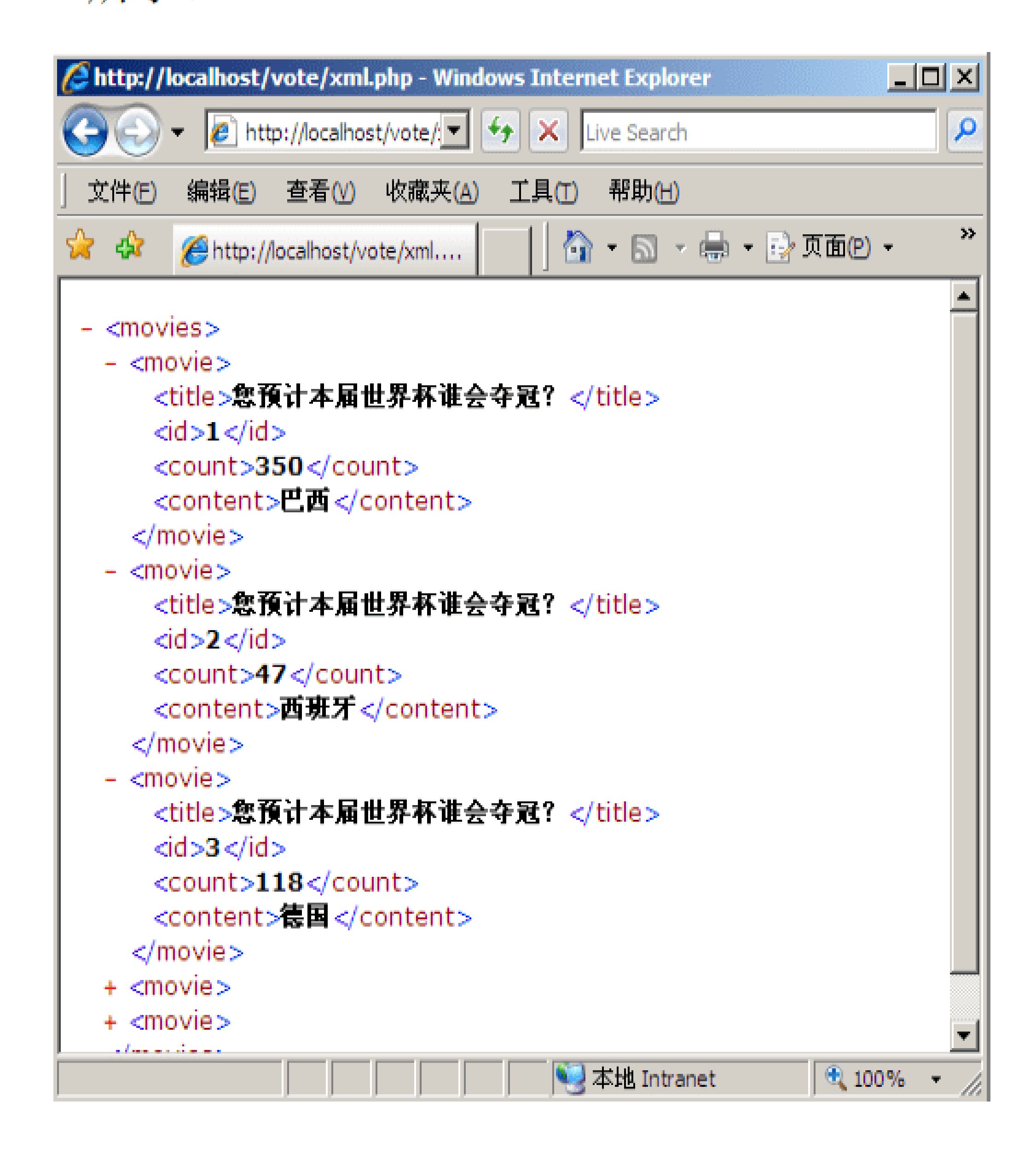


图 13.4　XML 格式的调查数据

从图 13.4 中可以看到，生成 XML 文档是标准的 XML 格式文件。在 PHP 中可以通过 XML 的解析函数来获取其中的信息，这里使用 simplexml_load_string()函数来获得，其功能是把 XML 字符串载入对象中。该函数如果失败，返回 false。具体的语法表述如下，其参数说明如表 13.2 所示。

```
simplexml_load_file(string,class,options,ns,is_prefix)
```

表 13.2　simplexml_load_string()函数参数含义表

参　数	描　述
string	必需。规定要使用的 XML 字符串
class	可选。规定新对象的 lass
options	可选。规定附加的 Libxml 参数
ns	可选
is_prefix	可选

该函数的返回值为返回类 SimpleXMLElement 的一个对象。该对象的属性包含 XML 文档中的数据，如果失败则返回 false。

获取 XML 格式的调查内容的代码如下：

```
//引入 xml 文档
Include(xml.php);
//解析 xml 格式调查结果
$xml = simplexml load string($xmlstr);
$count0=$xml->movie[0]->count;
Echo $count0;
```

上面代码的含义是，获得了第一个节点中的 count 标签中的数据，执行下面的代码输入值为：

```
350
```

（3）完成了上面的程序方面的准备工作后，创建 view.php 文件作为在线调查程序的程序脚本。在线调查程序的具体代码如下：

```
<?php
/********************************************
'文件名：view.php
'主要功能：在线调查显示程序
'说明：
 ********************************************/
require_once (dirname ( __FILE__ ) . "/dbconfig.php");//数据库配置文件
require once (dirname (  FILE  ) . "/include.functions.php");
                                                    //公共函数库文件
$mysql  =  mysql connect  (  $mysqlconf  ["host"],  $mysqlconf  ["user"],
$mysqlconf ["pass"] ) or die ( 'Not connected : ' . mysql error () );
mysql select db ( $mysqlconf ["db"], $mysql ) or die ( 'Can\'t use foo :
' . mysql_error () );
mysql query ( "SET NAMES utf8" );                   //设置默认字符集
$aid = $ GET ['aid'];                               //获得栏目 id
if (! empty ( $aid )) {                             //判断栏目是否存在
    $sql = "SELECT * FROM 'vote' WHERE 'aid' ='" . $aid . "'";
    $result = mysql query ( $sql );
    $arr = mysql_fetch_array ( $result );
    //载入调查内容 xml 格式
    $xmlstr = "<?xml version=\"1.0\" encoding=\"utf-8\" ?>";
    $xmlstr .= $arr ['votenote'];
    //解析 xml 格式调查结果
    $xml = simplexml load string ( $xmlstr );
} else {
    echo '<script type="text/javascript">alert(\'该调查异常\')</script>';
    exit ();
```

```
}
?>
```

【代码解读】

上述程序的作用是显示在线调查内容。创建数据库连接后从页面获得（Get 方式）栏目 id，在数据库中获得相应调查内容数组（XML 格式），使用 simplexml_load_string()函数解析数组在浏览器中输出显示内容。

（4）对应的调查页面显示模板代码如下：

```
<HTML xmlns="http://www.w3.org/1999/xhtml">
<HEAD>
<TITLE>在线调查</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="./nspage.css" type=text/css rel=stylesheet>
</HEAD>
<BODY>
<DIV class="main mceneter">
<DIV class=toplogo>
    <DIV class=title>在线调查 </DIV>
</DIV>
<table width='100%' border='0' cellspacing='1' cellpadding='1' bgcolor=
'#FFFFFF'>
<form name='voteform' method='post' action='vote.php' target=' blank'>
<!--发送调查栏目 id-->
<input type='hidden' name='aid' value='1'>
<tr align='center'>
    <td height='24' bgcolor='#EDEDE2' ><?php echo $arr['votename']; ?></td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
        <input type='radio' name='voteitem' value='0'>
        <?php echo $xml->movie[0]->content; ?>
    </td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
    <input type='radio' name='voteitem' value='1'>
    <?php echo $xml->movie[1]->content; ?>
    </td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
    <input type='radio' name='voteitem' value='2'>
    <?php echo $xml->movie[2]->content; ?>
    </td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
    <input type='radio' name='voteitem' value='3'>
    <?php echo $xml->movie[3]->content; ?>
    </td>
</tr>
<tr>
    <td height=24 bgcolor=#FFFFFF>
    <input type='radio' name='voteitem' value='4'>
    <?php echo $xml->movie[4]->content; ?>
    </td>
</tr>
<tr>
```

```
    <td height='24' bgcolor='#FFFFFF'>
    <input type='submit' style='width:40;background-color:#EDEDE2;border:
    1px soild #818279' name='vbt1' value='投票'>
    <input type='button' style='width:80;background-color:#EDEDE2;border:
    1px soild #818279' name='vbt2' value='查看结果' onClick="window.
    open('result.php?aid=1');">
</td>
</tr>
</form>
</table>
</DIV>
</BODY>
</HTML>
```

（5）在浏览器中执行链接：

```
http://localhost/vote/view.php?aid=1
```

预览效果如图 13.5 所示。

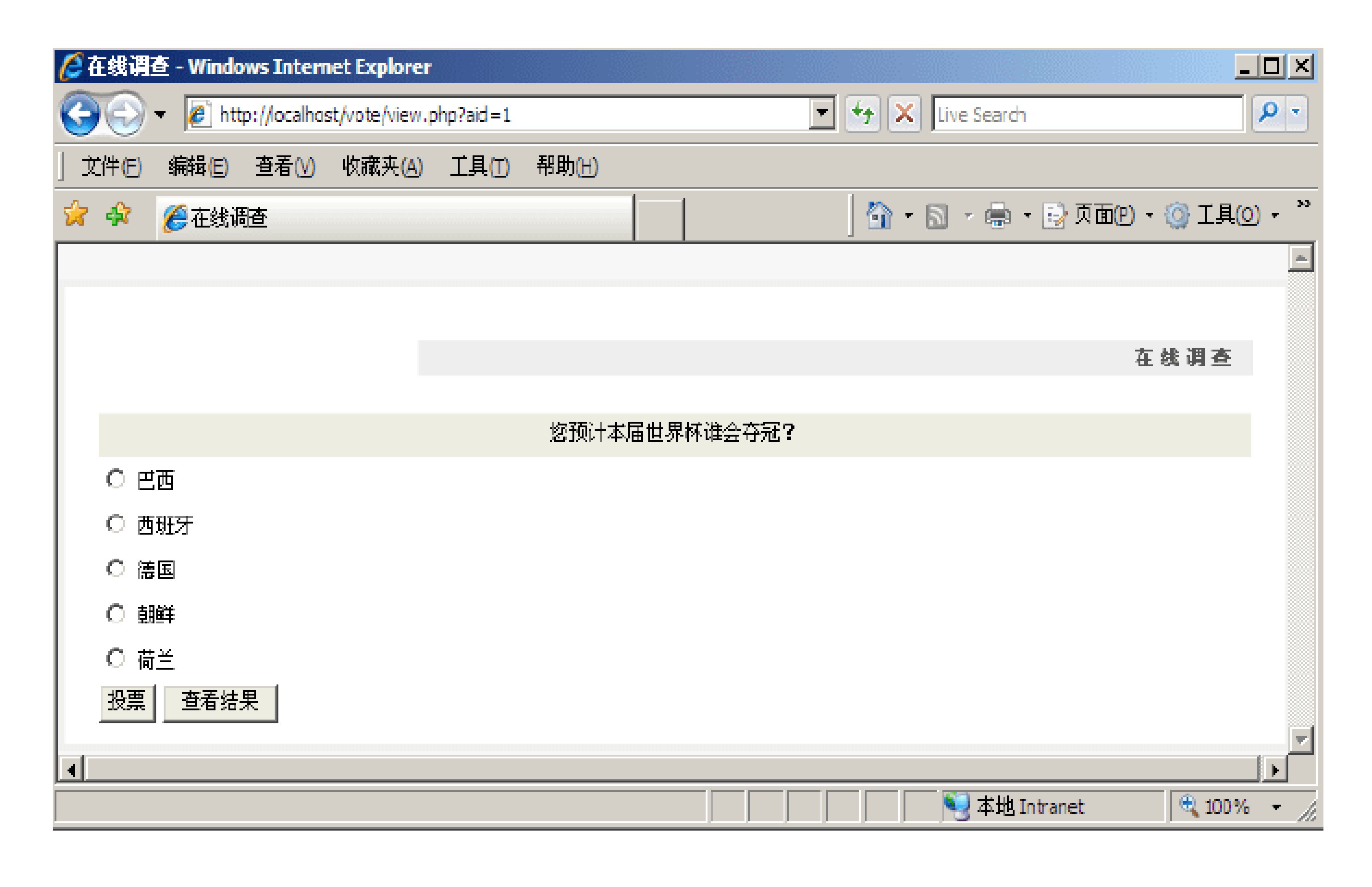


图 13.5　在线调查

13.5　调查结果查看

当访问者提交了调查数据后，希望可以了解调查结果（这是一个必要的互动步骤）。调查结果会以直观（如柱形图）的方式呈现给访问者，其中，还可以清楚地了解到总调查数及各个子选项的得票情况。

13.5.1　调查结果查看界面

调查结果页面的设计，包含了调查信息的展示和柱状图显示两个部分。创建文件名为 result.php 的 HTML 脚本，该文件将显示调查结果查看页面。文件具体代码设计如下：

```
<HTML xmlns="http://www.w3.org/1999/xhtml">
<HEAD>
<TITLE>投票结果</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
```

```
<LINK href="./nspage.css" type=text/css rel=stylesheet>
</HEAD>
<BODY>
<DIV class="main mceneter">
<DIV class=toplogo>
  <DIV class=title>在线调查 </DIV></DIV>
<DIV class=cmain>
<DIV class=ctitle>
<H1>调查结果</H1>
<SPAN></SPAN></DIV>
<DIV class="cbox mceneter">
<DIV class=votelist>
<DIV class=vtitle>
<H2>您预计本届世界杯谁会夺冠？</H2>
<SPAN>投票开始时间：0000-00-00   投票结束时间：2010-06-21  参与这次投票的总人数：
717</SPAN>
</DIV>
<DIV class=vbox>
<DL>
  <DT>1、巴西</DT>
  <DD><SPAN style="WIDTH: 49%"><STRONG>49%</STRONG>(350)</SPAN></DD>
</DL>
<DL>
  <DT>2、西班牙</DT>
  <DD><SPAN style="WIDTH: 7%"></SPAN><STRONG>7%</STRONG>(47)</DD>
</DL>
<DL>
  <DT>3、德国</DT>
  <DD><SPAN style="WIDTH: 16%"></SPAN><STRONG>16%</STRONG>(118)</DD>
</DL>
<DL>
  <DT>4、朝鲜</DT>
  <DD><SPAN style="WIDTH: 17%"></SPAN><STRONG>17%</STRONG>(122)</DD>
</DL>
<DL>
  <DT>5、荷兰</DT>
  <DD><SPAN style="WIDTH: 11%"></SPAN><STRONG>11%</STRONG>(80)</DD>
</DL>
  </DIV>  </DIV>  </DIV>  </DIV> </DIV>
</BODY>
</HTML>
```

【代码解读】

上述代码的核心是定义了如下的格式，显示柱状图和得票信息，代码样式如下：

```
<DD>
    SPAN style="WIDTH:得票百分数>">
    <STRONG>得票百分数</STRONG>(总票数)
    /SPAN>
</DD>
```

13.5.2　调查结果查看程序

通过获得调查主题编号，从数据库中获得该调查主题的全部信息，并将信息处理成格式化的标签输出在页面上。打开 result.php 文件改造该文件，加入 PHP 脚本处理逻辑，并替换 HTML 页面的信息和调查数据标签。具体的代码如下：

```
<?php
/*******************************************
'文件名：result.php
'主要功能：在线调查结果程序
'说明：
 *******************************************/
require once (dirname (   FILE   ) . "/dbconfig.php");//数据库配置文件
require once (dirname (   FILE   ) . "/include.functions.php");
                                               //公共函数库文件
$mysql  =  mysql connect  ( $mysqlconf  ["host"], $mysqlconf  ["user"],
$mysqlconf ["pass"] ) or die ( 'Not connected : ' . mysql error () );
mysql_select_db ( $mysqlconf ["db"], $mysql ) or die ( 'Can\'t use foo :
' . mysql error () );
mysql_query ( "SET NAMES utf8" );                         //设置默认字符集
$aid = $ GET ['aid'];                                     //获得栏目 id
if (! empty ( $aid )) {                                   //判断栏目是否存在
    $sql = "SELECT * FROM 'vote' WHERE 'aid' ='" . $aid . "'";
    $result = mysql_query ( $sql );
    $arr = mysql fetch array ( $result );
    //载入调查内容 xml 格式
    $xmlstr = "<?xml version=\"1.0\" encoding=\"utf-8\" ?>";
    $xmlstr .= $arr ['votenote'];
    $xml = simplexml_load_string ( $xmlstr );            //解析 xml 格式调查结果
    //获得结果所占小数
    $n1 = $xml->movie [0]->count / $arr ['totalcount'];
    $n2 = $xml->movie [1]->count / $arr ['totalcount'];
    $n3 = $xml->movie [2]->count / $arr ['totalcount'];
    $n4 = $xml->movie [3]->count / $arr ['totalcount'];
    $n5 = $xml->movie [4]->count / $arr ['totalcount'];
} else {
    echo '<script type="text/javascript">alert(\'该调查异常\')</script>';
    exit ();
}
?>
```

【代码解读】

上述代码的过程与前面的程序很相似。不同的地方是，在载入调查内容（XML 格式）数据后解析调查内容结果，然后计算结果所占的小数，生成序列化数组。

调查结果查看程序，对应的页面显示模板代码如下：

```
<HTML xmlns="http://www.w3.org/1999/xhtml">
<HEAD>
<TITLE>投票结果</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="./nspage.css" type=text/css rel=stylesheet>
</HEAD>
<BODY>
<DIV class="main mceneter">
<DIV class=toplogo>
  <DIV class=title>在线调查 </DIV></DIV>
<DIV class=cmain>
<DIV class=ctitle>
<H1>调查结果</H1>
<SPAN></SPAN></DIV>
<DIV class="cbox mceneter">
<DIV class=votelist>
<DIV class=vtitle>
```

```
<H2><?php echo $arr['votename']; ?></H2>
<SPAN>投票开始时间：<?php echo $arr['starttime']; ?>  投票结束时间：<?php echo
$arr['endtime']; ?>  参与这次投票的总人数：<?php echo $arr['totalcount']; ?>
</SPAN>
</DIV>
<DIV class=vbox>
<DL>
  <DT>1、<?php echo $xml->movie[0]->content; ?></DT>
    <DD>
        <SPAN style="WIDTH: <?php echo sprintf("%01.0f", $n1*100).'%';?>">
        </SPAN>
        <STRONG><?php echo sprintf("%01.0f", $n1*100).'%';?></STRONG>
        (<?php echo $xml->movie[0]->count; ?>)
    </DD>
</DL>
<DL>
  <DT>2、<?php echo $xml->movie[1]->content; ?></DT>
    <DD>
        <SPAN style="WIDTH: <?php echo sprintf("%01.0f", $n2*100).'%';?>">
        </SPAN>
        <STRONG><?php echo sprintf("%01.0f", $n2*100).'%';?></STRONG>
        (<?php echo $xml->movie[1]->count; ?>)
    </DD>
</DL>
<DL>
  <DT>3、<?php echo $xml->movie[2]->content; ?></DT>
    <DD>
        <SPAN style="WIDTH: <?php echo sprintf("%01.0f", $n3*100).'%';?>">
        </SPAN>
        <STRONG><?php echo sprintf("%01.0f", $n3*100).'%';?></STRONG>
        (<?php echo $xml->movie[2]->count; ?>)
    </DD>
</DL>
<DL>
  <DT>4、<?php echo $xml->movie[3]->content; ?></DT>
    <DD>
        <SPAN style="WIDTH: <?php echo sprintf("%01.0f", $n4*100).'%';?>">
        </SPAN>
        <STRONG><?php echo sprintf("%01.0f", $n4*100).'%';?></STRONG>
        (<?php echo $xml->movie[3]->count; ?>)
    </DD>
</DL>
<DL>
  <DT>5、<?php echo $xml->movie[4]->content; ?></DT>
    <DD>
        <SPAN style="WIDTH: <?php echo sprintf("%01.0f", $n5*100).'%';?>">
        </SPAN>
        <STRONG><?php echo sprintf("%01.0f", $n5*100).'%';?></STRONG>
        (<?php echo $xml->movie[4]->count; ?>)
    </DD>
</DL>
  </DIV>  </DIV>  </DIV>  </DIV> </DIV>
</BODY>
</HTML>
```

在浏览器中执行链接：

```
http://localhost/vote/result.php?aid=1
```

预览效果如图 13.6 所示。

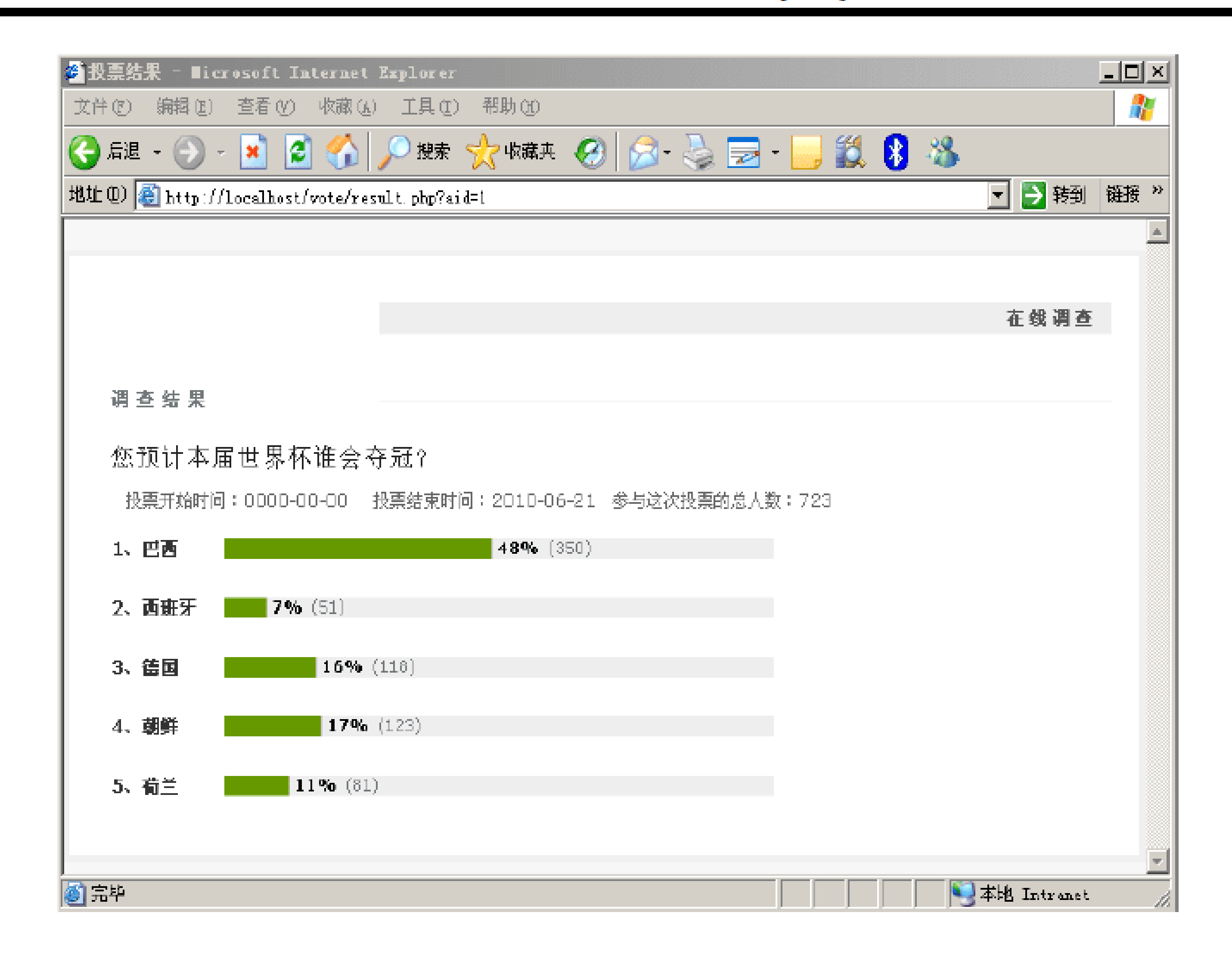


图 13.6　调查结果页面

13.6　小　　结

在线调查是指通过互联网及其调查系统，把传统的调查和分析方法在线化、智能化。本章的重点是介绍在线调查系统的基本原理，通过数据存储的中间载体（XML 文档）实现数据与显示内容的交互。在技术部分主要复习 PHP 脚本与 MySQL 数据库交互的相关内容、使用 XML 格式文档作为数据存储的中间载体实现数据存储，以及 HTML 页面标签实现数据的直观显示。

本章第 13.1 节介绍在线调查系统基本需求，通过图表的形式梳理了调查系统各个逻辑流程，使读者对在线调查系统的基本原理有了基本的了解。在第 13.2 节数据库设计部分通过需求分析，拆分出业务逻辑为建立调查存储表（vote），该表集合在线调查的全部数据内容。从第 13.3 节开始进入业务逻辑编程，介绍了调查管理后台程序。第 13.4 节是在线调查的显示逻辑程序，重点介绍页面标签与程序数据的交互，通过页面标签与程序数据的交互完成了在线调查模块的页面显示逻辑。第 13.5 节介绍了调查结果查看程序的编程，程序将数据库中获得的指定数据输出到页面模板中，完成调查结果查看程序。

第 14 章　WAP 资讯模块（PHP+WML+WAP）

WAP（Wireless Application Protocol，无线应用协议）最初由 WAP 论坛开发。通过该技术将无线移动通信技术和 Internet 结合起来，通过提供通用的平台，把目前 Internet 中广泛使用的 HTML（Hypertext Markup Language）语言描述的信息转换成用 WML（Wireless Markup Language）语言描述的信息，并在移动设备上显示。

本章主要介绍如何使用 PHP 构建一个典型的 WAP 应用。本章的学习目的是，了解 PHP 作为服务器脚本语言与 WML 的交互过程。在编程的方面，读者需要了解 WAP 应用的构成和标签规则。通过本章的介绍，读者可以使用本章模块提供的与基础构架扩展类似的基于 PHP 的 WAP 应用程序。

本章主要涉及的知识点如下。

- WAP 的定义：WAP（Wireless Application Protocol，无线应用协议），是一项全球性的网络通信协议。
- WML：WML（Wireless Makeup language，无线注标语言）WML 是以 XML 为基础的标记语言，被广泛地应用在 WAP 的页面显示中。
- error_log 日志记录函数：是 PHP 内置的日志函数包含 4 个参数（message 是消息内容、type 是消息类型、目标文件及其他的头信息）。
- urlencode 函数：是 PHP 内置的函数，用来将字符串以 URL 编码，例如，空格就会变成加号。

14.1　WAP 应用基础

WAP（Wireless Application Protocol，无线应用协议），是一项全球性的网络通信协议。WAP 使移动 Internet 有了一个通行的标准，其目标是将 Internet 的丰富信息及先进的业务引入到移动电话等无线终端之中。WAP 定义可通用的平台，把目前 Internet 网上 HTML 语言的信息转换成用 WML（Wireless Markup Language）描述的信息，显示在移动电话的显示屏上。由于静态的 WAP 页面在很多方面不能满足用户个性化的服务请求，因此通过 WAP 服务器端语言产生动态的 WML 页面是目前普遍采用的解决方法。

14.1.1　WAP 系统概述

WAP 应用结构非常类似于 Internet，其实，它就是一种 B/S 的模型执行过程，如图 14.1

所示。具体的执行流程如下。

（1）具有 WAP 用户代理功能的移动终端（WAP 手机等）通过内部运行的浏览器（这里的 Browser 不是传统意义上的浏览器，而是支持 WAP 设备上的用户代理，也可以理解为 WAP 浏览器），对某一网站以无线方式发送 WAP 服务请求。

（2）该请求先由 WAP 网关截获，对信息内容进行编码压缩，以减少网络数据流量。同时，根据需要将 WAP 协议转换成 HTTP 协议，然后将处理后的请求转送到相应 WAP 服务器。

（3）在 WAP 服务器端，分析文件扩展名选择相应的协议，被请求的页面直接或由服务器端脚本解释后输出，再经网关传回用户。

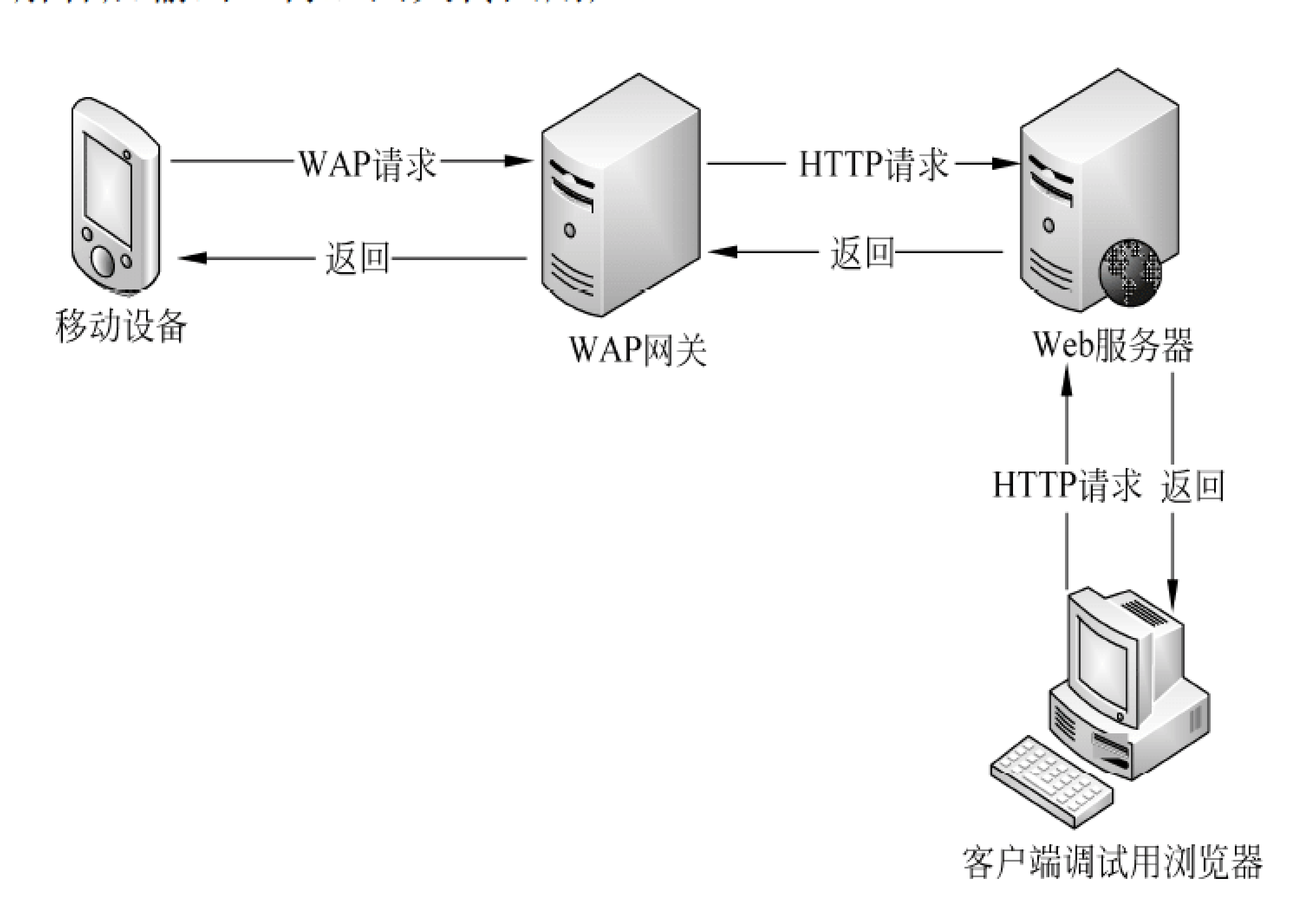


图 14.1　WAP 应用流程图

从上述 WAP 应用流程可以看到，生成动态 WAP 页面与动态产生 Web 网页的过程非常类似。但是，由于 WAP 应用使用的 WML 语言来源于语法严格的 XML，因此要求输出的格式必须按 WAP 网页的规范输出。同时，由于 WAP 协议的应用范围、移动客户端的软硬件水平等特殊性，对每次输出的页面的大小、图像的格式及容量都有一定的限制，因此通常使用服务器脚本语言动态输出 WAP 页面。

14.1.2　WML 基础

WML 是无线注标语言（Wireless Makeup language）的英文缩写。WML 是以 XML 为基础的标记语言，支持 WAP 技术的手机能浏览由 WML 描述的 Internet 内容。

WML 支持文字和图片显示。内容组织上，一个页面为一个 Card，而一组 Card 则构成一个 Deck。当使用者向服务器提出浏览要求后，WML 会将整个 Deck 发送至客户端的浏览器，使用者就可以浏览 Deck 里面所有 Card 的内容，而不需要从网络上单独下载每个 Card。

文档的实体包含在<wml></wml>标记中，文档里每个 CARD 又包含在<card></card>标记中，实际的内容标签包含在<wml></wml>标记之间。

请看一段 WXL 实例代码：

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml 1.1.xml">
<wml>
<card id="HELLO" title="HELLO">
Hello world!
</card>
</wml>
```

执行上面的代码显示结果如下：

```
Hello World!
```

enmWML 是 XML 的子集，继承了 XML 的字符集设置。WML 文档默认的字符集是 UTF-8，要显示其他字符集需要使用 encoding 属性。比如，要显示 GB2312 编码（简体中文）则需要将文档第一行改为：

```
<?xml version="1.0" encoding="gb2312"?>
```

WML 作为一种标记语言，作用是通过内置的各种标签创建可显示在 WAP 浏览器中的页面。在实际应用中，需要服务器端的脚本语言来动态将 WAP 站点中的内容生成 WML 标签输出到用户浏览器。下面将介绍如何使用 PHP 生成各种 WML 文档标签。

1．服务器设置

通常 WebServer 是默认支持 WML 标签的，如果服务器端不支持则需要修改 WebServer 的配置文件。在 Apache 中，需要 Apache 配置文件的 mime 属性，具体操作是在 conf/mime.types 文件中添加如下代码：

```
text/vnd.wap.wml wml
image/vnd.wap.wbmp wbmp
text/vnd.wap.wmls wmls
application/vnd.wap.wmlc wmlc
application/vnd.wap.wmlscriptc wmlsc
text/vnd.wap.wmlscript wsc
text/vnd.wap.wmlscript wmlscript
```

重启 Apache 的服务就完成了配置的修改。

2．PHP生成WML

PHP 生成 WML 页面需要使用 header()函数生成一个文件头信息，告诉浏览器该文件的格式类型，这样浏览器就能正确解析了。这里需要注意的是，由于是使用 PHP 做脚本语言来生成 WML 标签，该文件的扩展名应为.php，否则代码不能被 WebServer 正确地解析。下面是使用 PHP 生成 WML 简单例子。

```
<?php
header ( "Content-Type: text/vnd.wap.wml" );
echo "<?xml version=\"1.0\"?>\n"?>
<wml>
<card id="main" title="PHPWML">
<p>
<?php
```

```
    echo "hello，world\n"; //测试脚本输出
?>
</p>
</card>
</wml>
```

执行上面的代码，运行结果如下：

```
hello，world
```

注意：以上的代码需要在 WAP 模拟器中浏览，因为普通的浏览器中无法识别 WML 标签，甚至会被认为是错误下载。这是因为在程序开头就声明了该输出文档为 WML 类型，该类型只有 WAP 设备能够识别并解释。

3．用PHP动态生成图像

WAP 应用使用一种特殊黑白的图像格式 WBMP。在实际的制作中，需要用一些工具将已有图像转换成 WBMP 格式，然后在 WML 文档中使用。PHP 的 GD 库（版本 1.8 以上）已经提供了相应函数。

```
<?php
Header("Content-type: image/vnd.wap.wbmp");            //声明文件类型
Sim = ImageCreate ( 50, 50 );                          //设置图像大小
Swhite = ImageColorAllocate ( Sim, 255, 255, 255 );    //设置图像颜色
Sblack = ImageColorAllocate ( Sim, 0, 0, 0 );
ImageRectangle ( Sim, 5, 5, 20, 20, Sblack );          //绘制矩阵
ImageWBMP ( Sim );                 //以 WBMP 格式将图像输出到浏览器或文件
ImageDestroy ( Sim );              //释放与 image 关联的内存（销毁）
?>
```

【代码解读】

该文件将在 WAP 模拟器中显示一个黑色矩形框。

注意：使用 GD 的图像函数库，必须在 PHP 配置中加载 PHP_GD.DLL 库文件。

4．中文字符处理

WAP 作为一种全球应用，选择了 UNICODE 2.0 作为其标准字符集编码，以便能同时处理包括英文、中文、日文、法文等多种文字。中文字符集是 GB2312 编码，不同的编码不能相互解释。因此不在两种编码之间通过码表进行转换，就会出现解析中文字符乱码现象。在 PHP 中，有内置的编码转换的程序和函数，例如，在 PHP 中可以使用下面的代码进行 GBK 和 UTF-8 的转换。

```
$str=iconv("GBK", "UTF-8", $string);
```

目前，大多数 WAP 手机都是使用 UTF-8 编码的，也就是采用 UNICODE 来编码。如果我们直接在 WML 使用中文字符（GB2312 编码），将会产生乱码，手机用户无法识别。所以我们在输出中文之前，要使用程序或函数对中文进行 UNICODE 的编码。而在少数支持 GB2312 编码的手机或 WAP 终端设备中，我们可以在程序中定义好文档的内码类型后即

可直接正确显示中文。代码如下：

```
<?php
header('Content-type: text/vnd.wap.wml; charset=gb2312');
                                              //声明文件类型、设置字符集
echo ('<wml>');                               //输出标签元素
echo ('<card>');
echo ('<p>');
echo ('中文测试');
echo ('</p> ');
echo ('</card> ');
echo ('</wml>');
?>
```

WAP 模拟器中浏览上面的代码显示结果如下：

```
中文测试
```

14.1.3　WAP 浏览器安装与调试

Opera 浏览器是一款适用于各种平台、操作系统的浏览器。Opera 浏览器因为它的快速、小巧和比其他浏览器更佳的标准兼容性获得广泛应用。由于 Opera 对 WML 的良好支持，本章将使用浏览器作为测试 WAP 页面模拟器。安装调试步骤如下：

（1）通过 http://www.opera.com/下载最新版本的 Opera 浏览器。将安装文件下载到本地硬盘中，双击安装文件，首先进入安装语言选择，如图 14.2 所示。这里选择简体中文，单击“确定”按钮。

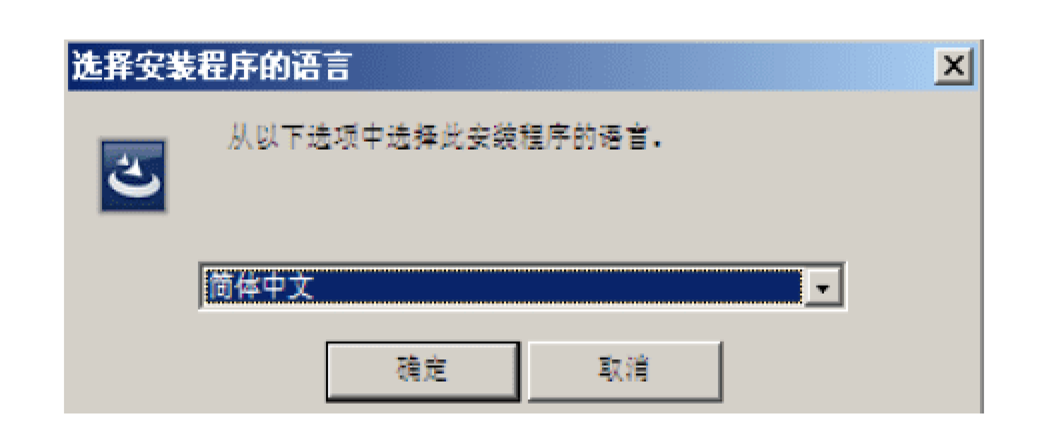


图 14.2　Opera 安装语言选择

（2）选择安装语言后会进入“安装向导”页面，这里默认选择“将 Opera 设置为默认浏览器”，建议取消勾选，如图 14.3 所示。

（3）单击“选项”按钮，页面如图 14.4 所示。在这里可以修改浏览器的安装路径，还有默认的创建图表关联，建议全部默认选中。



图 14.3　Opera 安装向导



图 14.4　Opera 安装选项

（4）设置完成后单击“关闭”按钮，返回“安装向导”页面，单击“下一步”按钮，进入浏览器许可协议页面，如图 14.5 所示，单击“同意”按钮。

安装向导会按照上面的设定，开始配置安装程序完成安装，由于默认选择了“安装完成后运行 Opera”，Opera 浏览器会在安装完成后自动运行。将前面介绍的 PHP 生成 WML 简单例子的代码保存为 test.php 文件，在 Opera 浏览器中执行此页面，如图 14.6 所示，证明 Opera 正确地解析了 WML 的标签。同样地，在手机和其他的移动设备上浏览 test.php 文件，页面的显示是完全一样的。

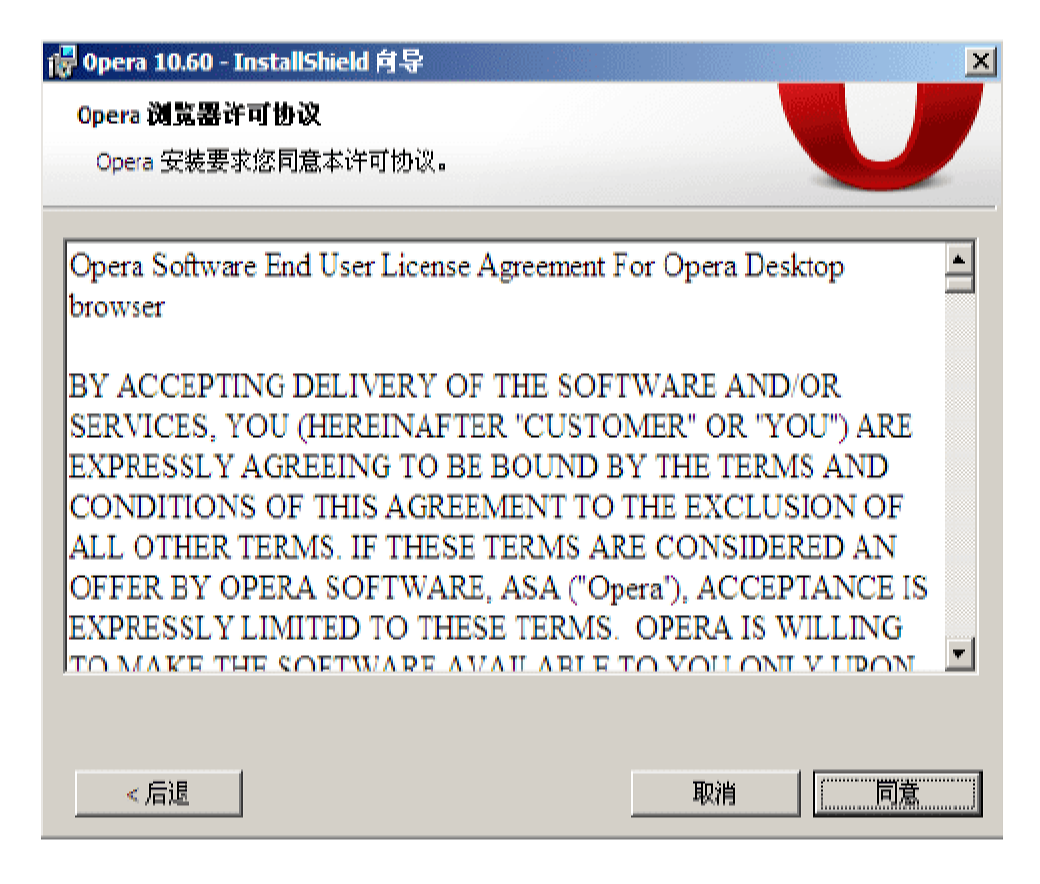


图 14.5　Opera 许可协议

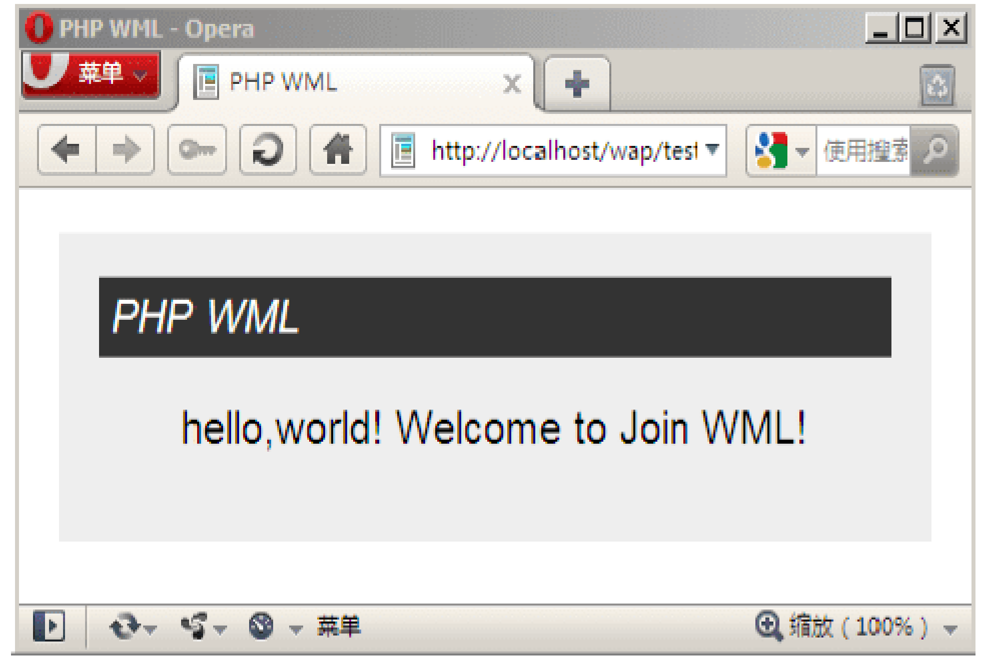


图 14.6　Opera 测试页面

14.2　数据库设计

WAP 资讯模块的设计目的是为了存储手机用户资料、为用户提供资讯服务。WAP 咨询数据库（wap）包含 3 个表，即用户资料表（userdb）、专辑资料表（albums）、图片资料库表（piclib）。下面来看下具体的设计。

14.2.1　用户资料表

用户资料表（userdb）的设计目的是记录手机用户的注册信息，信息包括两个部分：

- 个人资料（姓名、手机号、最后操作时间）。
- 账户资料（账户余额等）。

因为 WAP 应用经常涉及付费内容和身份确认的内容，所以在用户下载和浏览保密内容时，程序需要用到该表查询个人账户的信息和账户的余额。下面来看具体的设计步骤。

（1）建立 wap 数据库，SQL 语句如下：

```
CREATE DATABASE 'wap';
```

（2）在 wap 数据库中建立一个命名为 userdb 的用户资料表，结构如表 14.1 所示。创建表及相关字段的 SQL 语句如下：

```
CREATE TABLE 'userdb' (
  'uid' int(11) NOT NULL auto_increment,
```

```
  'username' varchar(32) default NULL,
  'mobile' char(11) NOT NULL,
  'endtime' datetime NOT NULL,
  'score' int(11) unsigned NOT NULL default '0',
  'note' varchar(256) NOT NULL,
  PRIMARY KEY  ('uid')
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;
```

表 14.1　用户资料表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
uid	int	11	Not null	auto_increment	自增主键
username	varchar	32	Null		用户名字
mobile	char	11	Not null		用户手机号
endtime	datetime		Not null		最后操作时间
Score	int	11	Not null	unsigned	账户余额
note	varchar	256	Not null		记录该次操作的备注信息

（3）为了方便程序调试，向 userdb 表中插入一条测试数据，语句如下：

```
--
-- 导出表中的数据 'userdb'
--
INSERT INTO 'userdb' VALUES (1, '中文名字', '13426210478', '2010-07-01
23:06:01', 4, '备注测试');
```

14.2.2　专辑资料表

专辑资料表（albums）的设计目的是记录专辑的信息，因为 WAP 资讯涉及的内容众多，而且多是以专辑的形式出现的，这样可以方便用户浏览提升用户体验。下面来看具体的设计步骤。

（1）在 wap 数据库中建立一个命名为 albums 的专辑资料表，结构如表 14.2 所示。创建表及相关字段的 SQL 语句如下：

```
CREATE TABLE 'albums' (
  'aid' int(11) NOT NULL auto increment,
  'catalogid' int(11) NOT NULL default '0',
  'title' varchar(255) default NULL,
  'desctext' text,
  'titlepic' varchar(255) NOT NULL default '',
  'createtime' datetime default NULL,
  'updatetime' datetime default NULL,
  'uid' int(11) NOT NULL default '0',
  'albumstatus' tinyint(1) NOT NULL default '1',
  'picnums' int(11) default '0',
  'views' int(11) default '0',
  'reviews' int(11) default '0',
  'scoretotal' int(11) default '0',
  'scoreperson' int(11) default '0',
  'position' text,
  'alt' varchar(40) default NULL,
  PRIMARY KEY  ('aid'),
  KEY 'catalogid idx' ('catalogid'),
  KEY 'picnums_idx' ('picnums'),
```

```
  KEY 'albumstatus idx' ('albumstatus'),
  KEY 'views idx' ('views')
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=22888 ;
```

表 14.2　专辑资料表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
aid	int	11	Not null	auto_increment	自增主键
catalogid	int	11	Not null		专题分类id
title	varchar	255	Null		专题名称
desctext	text				专题描述
titlepic	varchar	255	Not null		图片名称
createtime	datetime		Null		专题创建时间
updatetime	datetime		Null		专题更新时间
uid	int	11	Not null		专题所属的用户id
albumstatus	tinyint	1	Notnull		专题状态(1为发布、0为关闭)
picnums	int	11		0	专题中图片数目
views	int	11		0	专题浏览次数
reviews	int	11		0	专题评论数
scoretotal	int	11		0	专题消费数
scoreperson	int	11		0	专题消费人代码
position	text				专题中图片存储位置
alt					备注信息

（2）为了方便程序的调试，向 albums 表中插入一条测试数据，语句如下：

```
--
-- 导出表中的数据 'albums'
--
INSERT INTO 'albums' VALUES (22887, 0, '《黄金甲》', '巩姐、发哥。', '银幕搭档', '2010-07-01 10:51:19', '2010-07-01 10:51:23', 0, 1, 6, 100, 2, 20, 50, 'resize_images\\2006-12-18\\v', NULL);
```

14.2.3　图片资料库表

图片资料库表（piclib）的设计目的是存储专辑资料库中对应的图片详细资料，这些信息可以完整地描述图片的所有信息，如手机浏览 WAP 页面需要查看不同尺寸的图片信息。下面来看具体的设计步骤。

（1）在 wap 数据库中建立一个命名为 piclib 的“图片资料库表”，结构如表 14.3 所示。创建表及相关字段的 SQL 语句如下：

```
CREATE TABLE 'piclib' (
  'pid' int(11) NOT NULL auto increment,
  'aid' int(11) NOT NULL default '0',
  'desctext' text,
  'position' int(11) NOT NULL default '0',
  'picpath' varchar(255) NOT NULL default '',
  'picname' varchar(255) NOT NULL default '',
  'haveresize' tinyint(1) NOT NULL default '0',
```

```
  'have640' tinyint(1) NOT NULL default '0',
  'have800' tinyint(1) NOT NULL default '0',
  'have1024' tinyint(1) NOT NULL default '0',
  'createtime' datetime default NULL,
  'picstatus' tinyint(1) NOT NULL default '1',
  PRIMARY KEY  ('pid')
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=408242 ;
```

表 14.3　图片资料库表

字　段	数据类型	长　度	NULL	默认值	字段说明
pid	int	11	Not null	auto_increment	自增主键
aid	int	11	Not null		专题id
desctext	text				图片描述
position	int	11	Not null	0	图片位置信息
picpath	varchar	255	Not null		图片路径
picname	varchar	255	Not null		图片名称
haveresize	tinyint	1	Not null	0	是否有多种尺寸
have640	tinyint	1	Not null	0	小图片标识
have800	tinyint	1	Not null	0	中图片标识
have1024	tinyint	1	Not null	0	大图片标识
createtime	datetime		Null		创建时间
picstatus	tinyint	1	Not null	1	图片状态标识

（2）为了方便程序的调试，向 piclib 表中插入一组测试数据，语句如下：

```
--
-- 导出表中的数据 'piclib'
--
INSERT INTO 'piclib' VALUES (408239, 22887, '《黄金甲》描述 1', 0,
'resize images', 'LOG 15aehl.jpg', 0, 1, 1, 1, NULL, 1, 0);
INSERT INTO 'piclib' VALUES (408240, 22887, '《黄金甲》描述 2', 0,
'resize images', 'LOG 0xiz1v.jpg', 0, 1, 1, 1, NULL, 1, 0);
INSERT INTO 'piclib' VALUES (408241, 22887, '《黄金甲》描述 3', 0,
'resize_images', 'LOG_2cgjpr.jpg', 0, 1, 1, 1, NULL, 1, 0);
```

14.3　核心函数说明

通过前面的数据库设计和描述，相信读者已经对程序要实现的功能有了一个整体的了解了。在实际的开发过程中，为了提高代码的复用率和团队的开发效率，通常会将核心的功能函数预先写好保存在公共的函数库。这样，极大地方便了后续的业务程序的开发。创建一个 include.function.php 文件，用来保存 WAP 资讯模块的核心函数。

14.3.1　init_sid()函数

该函数的功能是，为当前运行的 Session 服务器变量给定一个 Session 编号 id。

具体的操作是先判断是否有传入的 sid 值（可以是从有合作关系的其他 WAP 站点跳转

过来的），如果有，设置$sid 变量；否则，使用 session_id()函数获得当前的 Session 编号 id 并赋值给$sid 变量。

函数代码如下：

```
function init sid()
{
    $sid = isset($ GET['sid'])?$ GET['sid']:'';    //检查变量是否存在
    if ($sid !='') session_id($sid);
    else
    {
        $sid = isset($_POST['sid'])?$_POST['sid']:'';
        if ($sid !='') session id($sid);
    }
    session_start();                               //初始化一个新的 Session
    $sid = session_id();                           //获得 Sessionid
    return $sid;
}
```

注意：session_id()函数是 PHP 的内置函数可取得或者重新配置目前存放 Session 的代号。若无参数 id，表示只有取得目前 Session 的代号；加上参数，则表示将 Session 代号设成新指定的 id。输入及返回均为字符串。

14.3.2　编码转换函数

编码转换函数一共有两个，即 gb2utf8()和 utf2gb()。gb2utf8()功能是将 GBK 的字符串转换成 UTF-8；utf2gb()是将 UTF-8 转换成 GBK 的字符串并返回处理结果。

具体的操作是使用 PHP 内置的编码转换函数 iconv()。

函数代码如下：

```
function gb2utf8($string)
{
    return iconv("GBK", "UTF-8", $string); //设置字符集转换
}

function utf2gb($string)
{
    return iconv("UTF-8", "GBK", $string); //设置字符集转换
}
```

这里需要注意的是，根据 PHP 手册中 iconv()的描述：

```
string iconv ( string in_charset, string out_charset, string str )
```

第一个参数是待转换的字符集编码，第二个参数是转换后的字符集编码。读者在使用的过程中注意不要混淆。

14.3.3　WML 格式输出函数

该函数的功能是为 PHP 生成的 WML 标签文档，输出一个 WML 格式的文件头，以便

于浏览器能够正确地识别后续的标签。

具体的操作是使用 PHP 内置的 header()函数，向浏览器输出文件头部信息，制定文件类型。

函数代码如下：

```
function output_waphead()
{
    header("Content-type: text/vnd.wap.wml;charset=UTF-8");
                                                    //声明文件类型、设置字符集
    header("Cache-Control: no-cache, must-revalidate");
    header("Pragma: no-cache");
    echo "<?xml version=\"1.0\"?>\r\n";            //向浏览器输出文件头
    echo "<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\"
    \"http://www.wapforum.org/DTD/wml 1.1.xml\">\r\n";
}
```

注意：PHP 中的 header()在使用时，它的上面不能够有任何的字符输出，否则程序会报错，不能正确执行。

14.4　WAP 认证程序

WAP 应用与普通的 Web 应用有所不同。由于用户使用的终端（PDA、手机等）都是有唯一标识的（比如手机号码等），因此在使用 WAP 应用和浏览 WAP 页面时，需要有认证的过程来确定访问者的身份信息。

14.4.1　登录认证程序

登录认证程序（login.php）是 WAP 程序认证部分的入口文件，后续的程序都是从这个文件作为入口的。程序的主要功能是通过 PHP 脚本生成一个手机号码验证的表单，下面介绍详细的代码实现。创建文件名为 login.php 的 PHP 脚本文件，代码如下：

```
<?
/*********************************************
'文件名：login.php
'主要功能：登录认证页面
'说明：
 ********************************************/
include ("include.function.php"); //加载函数库
output_waphead (); //输出文件头
?>
<wml>
<card title="login">
<p>
<?
//向浏览器输出内容标签
echo gb2utf8 ( "下载内容前，请输入您的手机号码进行验证：" );
echo "<br />\n<input name=\"mobile\" size=\"11\" maxlength=\"11\" format=
\"*m\" />\n";
```

```
echo "<anchor>" . gb2utf8 ( '确认' ) . "<go href=\"auth.php\" method=\
"get\">\n";
echo "<postfield name=\"mobile\" value=\"\$mobile\" />\n";
echo "<postfield name=\"sid\" value=\"" . $sid . "\" /></go></anchor>\n";
?>
</p>
</card>
</wml>
```

【代码解读】

上面的代码主要是 PHP 中的 echo 直接向浏览器输出 WML 标签，WML 的表单标签样式与 HTML 中的稍有不同，碰到陌生的标签读者可以查阅 WML 手册。在生成标签时，里面的标签值需要用“\”符号隔离，不然无法被 PHP 程序转译，实例样式如下：

```
<input name=\"mobile\" size=\"11\" maxlength=\"11\" format=\"*m\" />
```

在 Opera 浏览器中执行下面的链接：

```
http://localhost/wap/login.php
```

输入手机号后效果如图 14.7 所示。



图 14.7　登录认证页面

14.4.2　手机号码效验程序

通过登录认证程序（login.php）填入手机号码后，会发送表单信息到手机号码效验程序（auth.php）。程序的主要功能是对 login.php 提交过来的表单进行验证判断，根据表单提交的手机号检查数据库中是否有该条记录，如果有，则返回身份信息（账户余额等）。创建名为 auth.php 的 PHP 脚本文件，代码如下：

```
<?
/*******************************************
'文件名：auth.php
'主要功能：手机号码效验程序
'说明：
 *******************************************/
include ("include.function.php");                    //加载函数库
include ("include.mysqlconfig.php");                 //加载 MySQL 配置文件
$sid = init_sid ();                                  //获得 sid
$mobile = isset ( $_GET ['mobile'] ) ? $_GET ['mobile'] : '';
                                          //判断检查手机号码变量是否设置
$mysql = mysql_connect ( MYBILLHOST, MYBILLUSER, MYBILLPASS );
```

```
                                                    //创建数据库链接
mysql_select_db ( MYBILLDB, $mysql );
$_SESSION ['mobile'] = $mobile;                     //设置 Session 服务器变量
//验证手机号码 SQL 语句
$sql = "select * from userdb where mobile='" . $mobile . "' limit 1";
$result = mysql_query ( $sql, $mysql );
if ($row = mysql_fetch_array ( $result )) {
    $_SESSION ['score'] = $row ['score'];
                               //将获得账户中的余额设置为 Session 服务器变量
} else {
    $_SESSION ['score'] = 0;
}
mysql_close ( $mysql );
output_waphead ();
?>
<wml>
<card title="auth" ontimer="./bill.php?sid=<?=$sid?>">
<timer value="20" />
<p><a href="./bill.php?sid=<?=$sid?>"><?
echo gb2utf8 ( "认证成功自动跳转" );
?></a><br />
</p>
</card>
</wml>
```

【代码解读】

上面的代码主要是以提交的手机号变量$mobile为条件查询数据库的记录集，如果判断有返回值，则将账户的详细信息设置为全局的 Session 变量，后续的业务程序会预先判断这些身份变量。这里需要注意 WML 标签的跳转时间设置，代码如下：

```
<card title="auth" ontimer="./bill.php?sid=<?=$sid?>">
<timer value="20" />
```

Ontimer 中的值就是跳转的地址，<timer>标签中的 value 值就是跳转的等待时间。为了调试方便这里将时间设置较长（20），读者可以根据实际情况灵活调整。

登录认证程序（login.php）填入手机号码后，跳转到手机号码效验程序（auth.php），效果如图 14.8 所示。

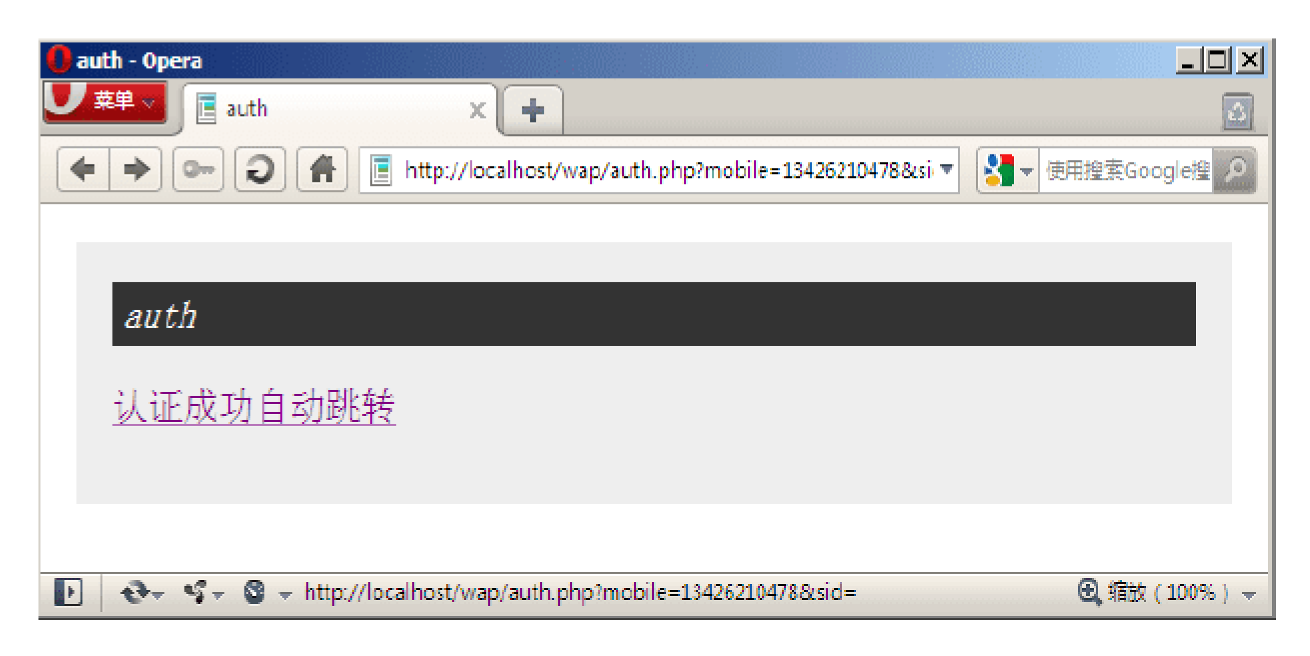


图 14.8　手机号码效验认证

14.4.3　付费内容认证程序

付费内容认证程序（bill.php），是认证部分的核心。程序的主要功能是设置效验全局

变量（包括操作内容、账户信息），如果条件验证满足，则对该次下载行为扣费，即对 wap 数据库中 userdb 表中的 score 字段进行扣费操作。

下面请看详细的代码实现。创建名为 auth.php 的 PHP 脚本文件，其中，付费内容认证逻辑部分的代码如下：

```
<?
/*******************************************
'文件名：bill.php
'主要功能：付费内容认证程序
'说明：
'*******************************************/
    include("include.function.php");        //加载函数库
    include("include.mysqlconfig.php");     //加载 MySQL 配置文件
    $sid = init_sid();                      //获得 Sid
    if (isset($_GET['backurl']))            //检查返回链接变量
    {
        $backurl = $_GET['backurl'];
        $_SESSION['backurl'] = $_GET['backurl'];
    }
    if (isset($_GET['downloadurl']))        //检查下载链接变量
    {
        $downloadurl = $_GET['downloadurl'];
        $_SESSION['downloadurl'] = $_GET['downloadurl'];
    }

    if (isset($_GET['downloadname']))       //检查下载名称变量
    {
        $downloadname = $_GET['downloadname'];
        $_SESSION['downloadname'] = $_GET['downloadname'];
    }

    if (isset($_SESSION['mobile']))         //检查手机号码变量
    {
        $mobile = $_SESSION['mobile'];
    }
    else
    {
        $_SESSION['mobile'] = '';
    }

    if (isset($_SESSION['score']))          //检查金额变量
    {
        $score = $_SESSION['score'];        //获得金额
    }
    else
    {
        $_SESSION['score'] = 0;             //余额归零
    }
    //创建 MySQL 连接
    $mysql = mysql_connect(MYBILLHOST, MYBILLUSER, MYBILLPASS);
        mysql_select_db(MYBILLDB, $mysql);
    output_waphead();
?>
```

【代码解读】

以上代码是判断各种认证参数的设置情况，同时加载对应的函数库、创建 MySQL 连

接，为认证过程作参数校验。

下面介绍付费内容认证程序的页面模板代码（WML 标签），具体的代码如下：

```
<wml>
<card title="bill" <?if ($_SESSION['score']> 0){?>ontimer="<?=$
downloadurl?>" <?}?>>
<p>
<?
if ($mobile == '') {
    echo gb2utf8 ( "下载内容前，请输入您的手机号码进行验证：" );
    echo "<br />\n<input name=\"mobile\" size=\"11\" maxlength=\"11\"
    format=\"*m\" />\n";
    echo "<anchor>" . gb2utf8 ( '确认' ) . "<go href=\"auth.php\" method=
    \"get\">\n";
    echo "<postfield name=\"mobile\" value=\"\$mobile\" />\n";
    echo "<postfield name=\"sid\" value=\"" . $sid . "\" /></go>
    </anchor>\n";
} else {
    if ($score > 0) {
        //下载扣费操作
        $_SESSION ['score'] = $_SESSION ['score'] - 1;
        $sql = "update userdb set score = score - 1, endtime= now() where
        mobile='" . $_SESSION ['mobile'] . "'";
        $result = mysql_query ( $sql, $mysql );
        if (! empty ( $result )) {
            //下载链接
            echo "<timer value=\"100\" />";
            echo "<a href=\"" . $downloadurl . "\">" . gb2utf8("点击下载").
            "</a><br /><br />\n";
        } else {
            //操作失败返回
            echo "<card title=\"auth\" ontimer=\"" . $backurl . "\">";
            echo "<timer value=\"5\" />";
            echo "<a href=\"" . $downloadurl . "\">" .gb2utf8 ( "操作失败") .
            "</a><br /><br />\n";
        }
    } else {
        //数据临时存放，也可以进行充值的数据库操作，这里略过
        //$sql='...';
        if (! empty ( $result )) {
        }
        //操作记录日志
        //error_log($_SESSION['alliance_id']."|".$_SESSION['mobile'].
        "|".date("Y-m-d H:i:s")."|".$sid."|".$_SESSION['downloadurl'].
        "\n", 3, "./wap/logs/bill_log.".date("Ymd", mktime()));
        //显示提示
        echo gb2utf8 ( "温馨提示：" ) . "<br />\n";
        echo gb2utf8 ( "系统已经缓存了您的内容《" );
        echo gb2utf8 ( $downloadname );
        echo gb2utf8 ( "》。" );
        echo "<br />" . gb2utf8 ( "请使用手机：" . $mobile . "发送短信" );
        if ((substr ( $mobile, 0, 3 ) >= '134' && substr ( $mobile, 0, 3 )
        <= '139') || substr ( $mobile, 0, 3 ) == '159') {
            echo gb2utf8 ( "编辑内容 HAHA 至 588518" ) . "<br />\n";
            echo gb2utf8 ( "（1.5 元/条，不含通讯费）" ) . "<br /><br />\n";
        } elseif (substr ( $mobile, 0, 3 ) >= '130' && substr ( $mobile, 0,
        3 ) <= '133') {
            echo gb2utf8 ( "编辑内容 X 至 9022980" ) . "<br />\n";
```

```
            echo gb2utf8 ( "（2 元/条，不含通讯费）" ) . "<br /><br />\n";
        }
        echo "<br /><br />\n";
        echo gb2utf8 ( "或者重新输入手机号码进行验证：" );
        echo "<br />\n<input name=\"mobile\" size=\"11\" maxlength=\"11\"
        format=\"*m\" />\n";
        echo "<anchor>" . gb2utf8 ( '确认' ) . "<go href=\"auth.php\
        "method=\"get\">\n";
        echo "<postfield name=\"mobile\" value=\"\$mobile\" />\n";
        echo "<postfield name=\"sid\" value=\"" . $sid . "\" /></go>
        </anchor><br />\n";
        echo gb2utf8 ( "客服电话：" ) . "010-1234567";
        echo "\n";
    }
}
mysql_close ( $mysql );
?>
</p>
</card>
</wml>
```

【代码解读】

由于在前面设置了全局的 Session 变量，因此后续的程序都可以使用这些变量控制操作流程和处理账户信息。这里通过$_SESSION['score']变量来控制了程序的流程，如果该值大于 0，说明账户有足够的余额可以进行扣费，程序会返回到真实的程序下载链接让用户下载，代码如下：

```
<card title="bill" <?if ($_SESSION['score']>0){?> ontimer="<?=
$downloadurl?>"<?}?>>
```

在 Opera 浏览器中执行付费内容认证程序（bill.php），页面预览如图 14.9 所示。在图中可以看到，显示了正确的下载地址链接给用户。

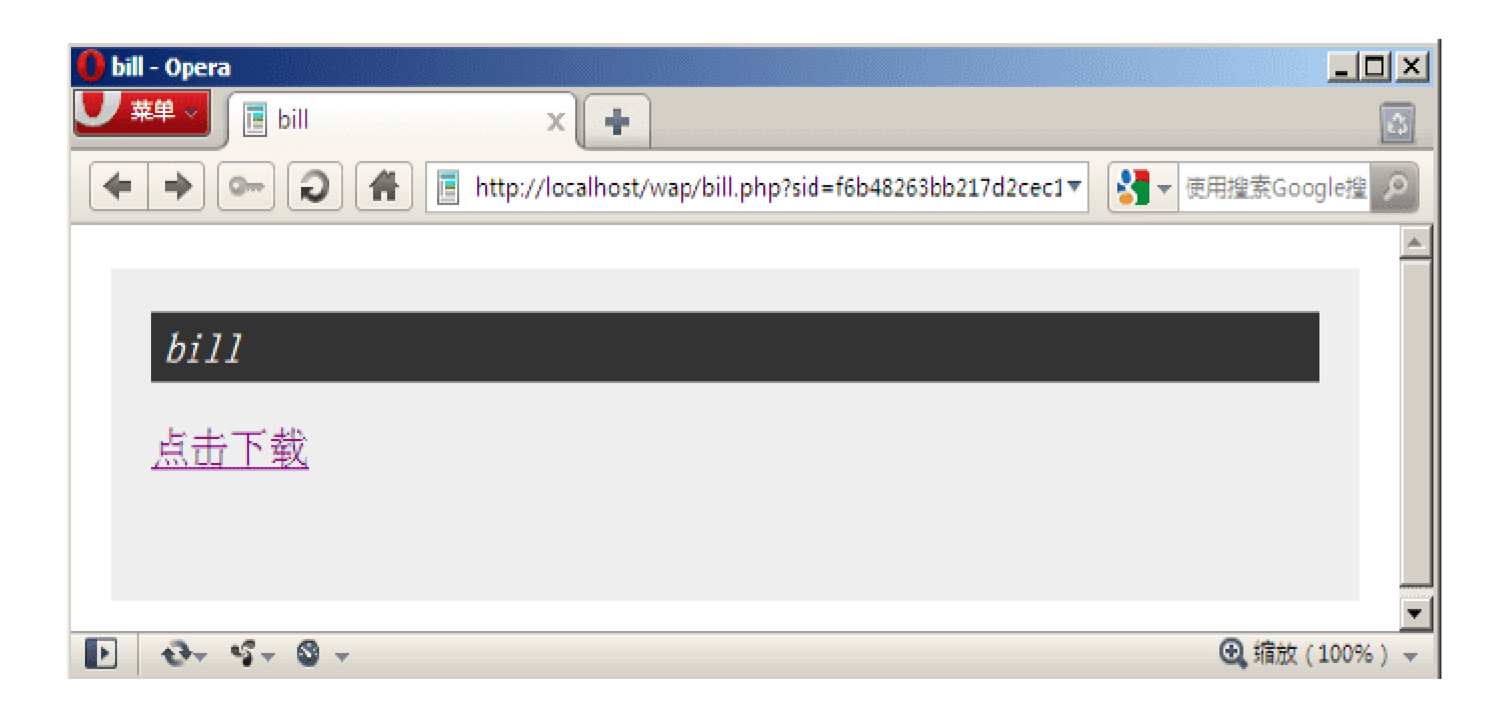


图 14.9　图片下载页面

如果余额不足则提示用户进行“充值操作”，系统会自动缓存该图片，待用户充值后显示正确的下载链接并记录该次用户的操作日志。PHP 有专门处理日志的函数 error_log()，用户记录该次操作的代码如下：

```
//error_log($_SESSION['alliance_id']."|".$_SESSION['mobile']."|".date("
Y-m-d H:i:s")."|".$sid."|".$_SESSION['downloadurl']."\n", 3, "./wap/logs/
bill_log.".date("Ymd", mktime()));
```

当该用户账户余额不足时，效果如图 14.10 所示，在图中可以看到，提示用户进行充

值操作，或者重新输入手机号码验证。

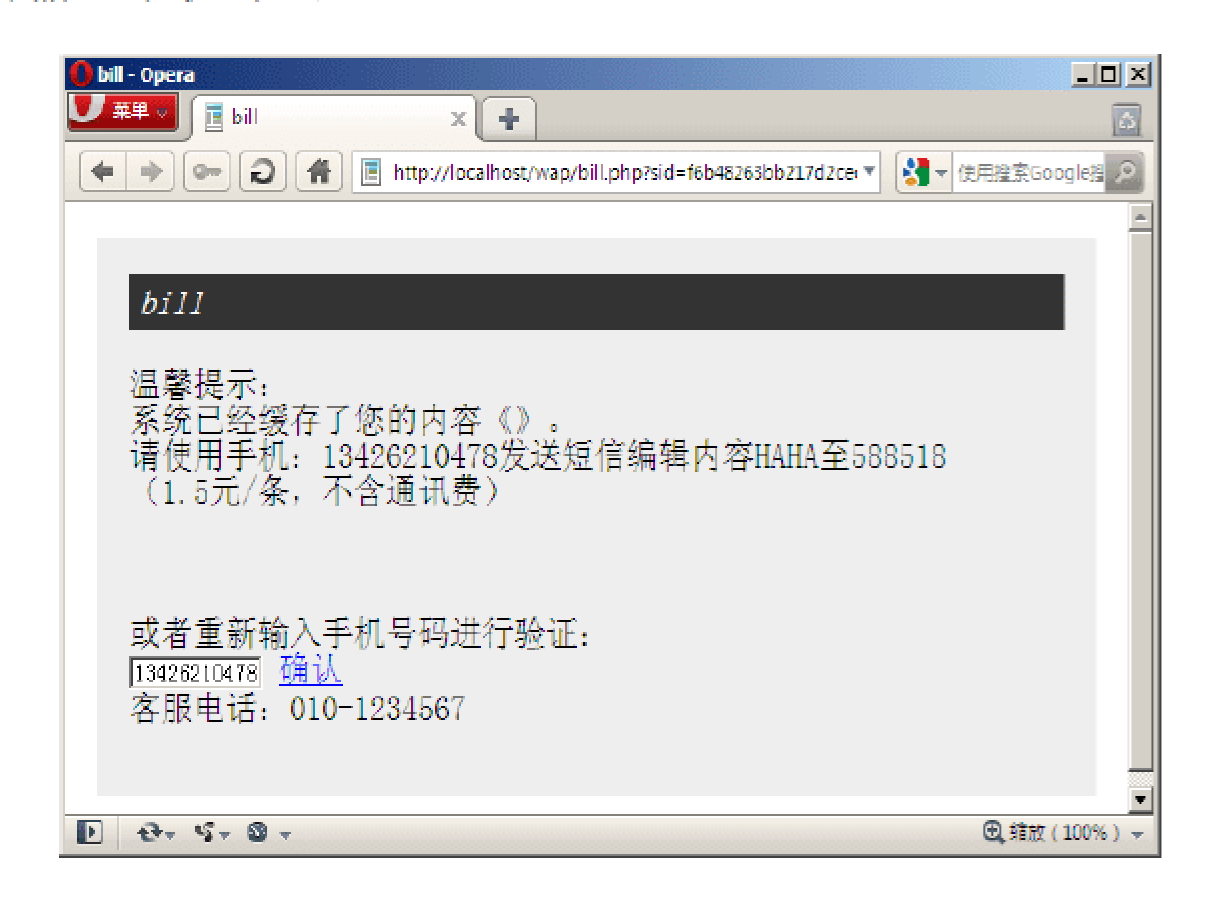


图 14.10　余额不足提示充值

14.5　WAP 资讯程序

WAP 资讯程序是提供给手机用户浏览的页面程序，由 WAP 首页程序（index.php）、专辑列表（list.php）程序、专辑图片显示（album.php）、图片显示程序（showpic.php）和图像下载程序（download.php）组成。其中，WAP 首页程序是用户浏览 WAP 的入口程序。

14.5.1　WAP 资讯首页程序

WAP 资讯首页程序是手机用户浏览站点最先访问的页面程序。它的特点是页面信息量大、便于用户按照栏目分类查找到需要的内容，同时这个页面也是更新最为频繁的页面，这样才能保证用户的良好体验。

WAP 资讯首页程序（index.php）是 WAP 资讯模块的核心程序，用户也是根据这个程序的导航分流到不同的栏目页面。程序的执行流程是：

（1）加载“Wap 首页推荐内容”配置文件，生成对应的 Smarty 标签；

（2）生成 WML 标签输出到 Opera 浏览器显示出来。

可能会有读者很疑惑，为什么不通过遍历 MySQL 数据库将对应栏目的内容取出来，而是以手动修改配置文件的方式更新首页内容？这是由首页内容的特点决定的，为了保证首页资讯的“新鲜”，首页中的内容更新频率非常高，而且每次推荐的规则也是由编辑根据当时的情况制定，所以程序就需要经常调整显示逻辑的部分。使用推荐配置文件就可以最大程度上减少程序员的工作量，同时也保证了更新的灵活，这个特点读者可以在实际操作中仔细体会。

下面请看详细的代码实现。创建文件名为 index.php 的 PHP 脚本文件，代码如下：

```
<?php
/*******************************************
'文件名：index.php
```

```
'主要功能：Wap 首页
'说明：
 *******************************************/
include_once (dirname ( __FILE__ ) . "/include.smarty.php");
                                            //加载 Smarty 配置文件
include (dirname ( __FILE__ ) . "/inc/include.function.php");
                                            //加载函数库文件
include ("inc/messagelist.php");            //加载推荐内容配置文件
output_waphead2 ();
$sid = init_sid ();                         //获得 sid
$debug = var_process ( "get", "debug", 0 );
//根据 debug 的值控制程序流程默认为 0
if ($debug == 1) {
    $str = "";
    $str = "<a href=\"/stat.php\">" . gb2utf8 ( "数据统计" ) . "</a><br />";
    $str .= "<a href=\"/clear.php?sid=" . $sid . "\">" . gb2utf8 ( "计费
    测试" ) . "</a><br />";
    $str .= "<a href=\"/.../?sid=" . $sid . "\">" . gb2utf8 ( "新版 WAP 首
    页" ) . "</a><br />";
    $str .= "<a href=\"/.../?sid=" . $sid . "\">" . gb2utf8 ( "WAP 推广页" ) .
    "</a><br />";
    $str .= "<a href=\"test.mp4\">test</a><br />";

    //print_r($_SESSION);
}
//导航部分
$smarty->assign ( "title", gb2utf8 ( "Wap 资讯手机版" ) );
$smarty->assign ( "ftvTitle", gb2utf8 ( "ftv 资讯" ) );
$smarty->assign ( "photoTitle", gb2utf8 ( "图吧" ) );
$smarty->assign ( "morephoto", gb2utf8 ( "更多美图…" ) );
$smarty->assign ( "moreftv", gb2utf8 ( "更多资讯…" ) );
//导航标签
$smarty->assign ( "sid", $sid );
$smarty->assign ( "ftvList", $ftvlist );
$smarty->assign ( "photoList", $photolist );
//ftv 资讯标签
$smarty->assign ( "ftvlist_imgurl", $ftvlist_imgurl );
$smarty->assign ( "ftvlist_imgId", $ftvlist_imgId );
$smarty->assign ( "ftvlist_img_title", $ftvlist_img_title );
//图吧显示标签 1
$smarty->assign ( "photolist_imgurl", $photolist_imgurl );
$smarty->assign ( "photolist_imgId", $photolist_imgId );
$smarty->assign ( "photolist_img_title", $photolist_img_title );
//图吧显示标签 2
$smarty->assign ( "photolist_img2url", $photolist_img2url );
$smarty->assign ( "photolist_img2Id", $photolist_img2Id );
$smarty->assign ( "photolist_img2_title", $photolist_img2_title );
$smarty->assign ( "str", $str );
//加载首页模板
$smarty->display ( "./index.tpl" );
?>
```

【代码解读】

根据 debug 变量的值控制程序的执行流程，其中变量值为 1 时，提供给编程及管理员获得统计数据和新功能测试时使用。

由于首页页面的信息量是最大页面，也是最为复杂的，因此直接使用 PHP 输出 WML 标签会造成程序“可读性”的严重下降，所以在程序中引入 Smarty 模板引擎，将程序代码与页面显示逻辑有效分离。下面请看首页模板部分的设计（index.tpl），代码如下：

```
<?xml version='1.0' encoding='utf-8'?><wml>
<card title="{|$title|}">
<p>
<br />
<a href="/ftv/?sid={|$sid|}">{|$ftvTitle|}</a>|
<a href="/photo/?sid={|$sid|}">{|$photoTitle|}</a>|
<a href="/game/">{|$gameTitle|}</a><br />
<b>手机游戏</b><br />
<a href="http://localhost/wap/game/2/">【新游】美少女梦工场之数独</a><br />
<a href="http://localhost/wap/game/1/">【新游】名侦探科南 精华版</a><br />
<b>{|$ftvTitle|}</b><br />
<a href="/ftv/detail.php?id={|$ftvlist imgId|}&amp;sid={|$sid|}">
<img src="{|$ftvlist imgurl|}" alt="{|$ftvlist img title|}" /></a><br />
{|foreach from=$ftvList item=ftvList|}
    <a href="/ftv/detail.php?id={|$ftvList. id|}&amp;sid={|$sid|}">
    {|$ftvList.title|}</a><br />
{|/foreach|}
<a href="/ftv/?sid={|$sid|}">{|$moreftv|}</a><br />
<b>{|$photoTitle|}</b><br />
<a href="/wap/photo/album.php?aid={|$photolist imgId|}&amp;sid=
{|$sid|}">
<img src="{|$photolist imgurl|}" alt="{|$photolist img title|}" /></a>
<a href="/wap/photo/album.php?aid={|$photolist  img2Id|}&amp; sid=
{|$sid|}">
<img src="{|$photolist img2url|}" alt="{|$photolist img2 title|}" />
</a><br />
 <!--循环读取数组-->
{|foreach from=$photoList item=photoList|}
<a href="/wap/photo/album.php?aid={|$photoList.id|}&amp;sid={|$sid|}">
{|$photoList.title|}</a>
<br />
{|/foreach|}
<a href="">{|$morephoto|}</a><br />
{|$str|}
<a href="">{|$ftvTitle|}</a>|
<a href="">{|$photoTitle|}</a>|
<a href="">{|$gameTitle|}</a>
<br />
<b>【友情链接】</b><br />
<a href="http://3gyy.net.cn/?fr=ifensi">【名称】YY 搜索</a><br />
<a href="http://wap.vmud.net/">【名称】网灵空间(免费)</a><br />
<a href="http://52d.cc/dodomusic/dodomiusic.jsp?fid=455">【名称】dodo 手机
音乐在线</a><br />
<a href="http://mo.yqmedia.com/e.jsp?w=1027">【名称】软件下载</a><br />
</p>
</card>
</wml>
```

现在打开首页推荐内容配置文件（messagelist.php），看资讯数组是如何生成的，代码如下：

```
<?php
/*********************************************
'文件名：messagelist.php
```

```
'主要功能：Wap 首页推荐内容
'说明：
'*******************************************/
//WAP 首页显示内容
$ftvlist imgurl = "img/ftv01.jpg";
$ftvlist_imgId  = 97108;
$ftvlist img title  = gb2utf8("世界杯变身“欧洲悲”");
//ftv 资讯标题
$ftvlist[0]['title']  = gb2utf8("富士康生产线将迁往河北廊坊");
$ftvlist[1]['title']  = gb2utf8("珠海破获 40 亿元网络赌博团伙");
$ftvlist[2]['title']  = gb2utf8("欧盟对我数据卡发反倾销调查");
$ftvlist[3]['title']  = gb2utf8("内蒙大兴安岭林区火点扑灭");
$ftvlist[4]['title']  = gb2utf8("吉林企业操纵绿豆价被罚 100 万");
//ftv 资讯对应的文章 id
$ftvlist[0]['id']  = 95961;
$ftvlist[1]['id']  = 97093;
$ftvlist[2]['id']  = 96881;
$ftvlist[3]['id']  = 96195;
$ftvlist[4]['id']  = 96775;
//图吧推荐图片
$photolist imgurl = "img/2.gif";
$photolist imgId    = 22887;
$photolist img title    = gb2utf8("《黄金甲海报 1》");
$photolist img2url = "img/3.gif";
$photolist img2Id   = 22887;
$photolist_img2_title   = gb2utf8("《黄金甲海报 2》");
//图吧推荐图片名称
$photolist[0]['title']  = gb2utf8("时尚派对异域花妖装");
$photolist[1]['title']  = gb2utf8("漂亮女大学生俏皮自拍");
$photolist[2]['title']  = gb2utf8("新世纪福音战士组图");
$photolist[3]['title']  = gb2utf8("名侦探柯南组图");
$photolist[4]['title']  = gb2utf8("人型电脑天使心组图");
//图吧推荐图片 id
$photolist[0]['id']  = 8322;
$photolist[1]['id']  = 8340;
$photolist[2]['id']  = 8345;
$photolist[3]['id']  = 8328;
$photolist[4]['id']  = 8338;
?>
```

【代码解读】

在上面的代码中，定义以下几类变量：

- 标题变量。
- 标题对应的 id 变量。
- 推荐图片变量。

注意：标题变量的个数和顺序要与标题 id 变量相对应，否则程序会出现对应错误的情况。

在 Opera 浏览器中执行下面的链接：

```
http://localhost/wap/index.php
```

输入手机号后效果如图 14.11 所示。



图 14.11　WAP 资讯首页

14.5.2　专辑图片

专辑图片（album.php）程序其实是一个内容显示程序，用户在 WAP 首页选择需要浏览的链接就进入了内容显示的页面。这里以“图吧”的专辑显示程序为例，说明如何实现内容显示，ftv 资讯频道的内容显示原理也是一样的，请读者在学习完后试着独立完成。

下面请看详细的代码实现。创建文件名为 album.php 的 PHP 脚本文件，代码如下：

```
<?
/*******************************************
'文件名：album.php
'主要功能：图吧专辑图片
'说明：
'*******************************************/
    include("include.mysql.php");                  //加载 MySQL 连接配置
    include("include.function.php");               //加载函数库
    include("class.album.php");                    //加载图片处理类
    $sid = init_sid();                             //获得 sid
    $aid = var_process("get", "aid", 7181);        //获得 aid
    $currpage = var_process("get", "page", 1);
    $error_output = "";                            //初始化报错信息
    $album = new Album($aid, $mysql);              //初始化图片类
    if (!$album->get($aid, $mysql))
    {
        $error_output = "参数错误！2";
    }
    else if ($album->albumStatus != 1)
    {
        $error_output = "参数错误！3";
    }
    else if (!$album->position || !$album->picNums || !$album->title)
    {
        $error_output = "参数错误！4";
    }
    else
    {
        $picInfo = $album->getPicList($mysql); //获得专辑中图片的 list
        $picsum = count($picInfo);                 //当图片很多时生成分页
        $pageInfo = divpage_toadmin($picsum, $currpage, 5);
        if ($currpage <= 1)                        //判断当前页码
        {
            $divpage = "<br />";
            $currpage = 1;
        }
        else
        {
            $divpage = "<br /><a href=\"album.php?sid=".$sid."&amp;
            aid=".$aid."&amp;page=".($currpage-1)."\">".gb2utf8("看看前面
            的")."</a> ";
        }
        if ($currpage >= $pageInfo["max_page"])
        {
            $divpage .= "<br />\n";
            $currpage = $pageInfo["max_page"];
        }
```

```
        else
        {
            $divpage.= "<a href=\"album.php?sid=".$sid."&amp;aid=".
            $aid."&amp;page=".($currpage+1)."\">".gb2utf8("更多图片... ").
            "</a><br />\n";
        }
        //计算分页数目
        $n = ($currpage-1)*5;
        for ($i=0; $i < 5;$i++)
        {
            if (!isset($picInfo[$n])) break;
            $piclog[$i]["picurl"] = "".$picInfo[$n]["picpath"]."/".
            $picInfo[$n]["picname"];
            $piclog[$i]["pid"] = $picInfo[$n]["pid"];
            process titleimage($picInfo[$n]["picpath"]."/".$picInfo[$n]
            ["picname"]);
            $n++;
        }
        $_SESSION['downloadname'] = $album->title;     //获得下载名称变量
    }
    mysql_close($mysql);                               //关闭数据库连接
    output waphead();
?>
<wml>
<card title="album pic list">
    <p>
<?php
if ($error_output != "") {
    echo "<p>".gb2utf8($error output)."</p>\n";        //编码转换
    } else {
    echo gb2utf8($album->title)."<br />\n";            //输出标题内容
    for ($i=0; $i< count($piclog);$i++)
    {
echo "<img src=\"".$piclog[$i]["picurl"]."\" alt=\"".$i."\" /><br />\n";
echo "<a href=\"showpic.php?sid=".$sid."&amp;aid=".$aid."&amp;pid=
".$piclog[$i]['pid']."\">".gb2utf8("大图")."</a> ";
echo "<a href=\"showpic.php?sid=".$sid."&amp;aid=".$aid."&amp;pid=
".$piclog[$i]['pid']."\">".gb2utf8("下载")."</a><br />\n";
    }
    echo $divpage."\n";
}
?>
<br />
<a href="list.php?sid=<?=$sid?>&amp;catalogid=<?=$album->catelogID?>"> <?
echo gb2utf8("返回列表页");?></a><br />
<a href="/photo/?sid=<?=$sid?>"><?echo gb2utf8("返回首页");?></a>
</p>
</card>
</wml>
```

【代码解读】

上面的代码中，为了处理图片专辑，创建了一个专门处理图片转进的 Album 图片类，应用里面的 get()方法来判断图片专辑的完整性，使用 getPicList()获得专辑中图片的 list。由于有些专辑中图片过多，手机的显示限制需要分页显示，因此对获得的图片做了分页处理，读者在调试程序时可以仔细体会。

在 Opera 浏览器中执行下面的链接：

```
http://localhost/wap/photo/album.php?aid=22887&sid=f6b48263bb217d2cec1d-
72ceb84ae6e5
```

专辑图片的效果如图 14.12 所示。

图 14.12　专辑图片

14.5.3　专辑列表

专辑列表（list.php）的设计目的是，用户在该页面可以通过列表选择图片的专辑。在编码实现方面与专辑图片程序非常相似，也是使用 Album 图片类，判断图片的类型并将获得的结果数组遍历出来，输出到浏览器中。

下面请看详细的代码实现。创建文件名为 list.php 的 PHP 脚本文件，代码如下：

```
<?
/*******************************************
'文件名：list.php
'主要功能：专辑列表
'说明：
'*******************************************/
    include("include.mysql.php");
    //加载 MySQL 连接配置
    include("include.function.php");                    //加载函数库
    include("class.album.php");                         //加载图片处理类
    $sid = init_sid();                                  //获得 sid
    $catalogid = var_process("get", "catalogid", 1);    //获得分类 id
    $currpage = var process("get", "page", 1);          //获得当前页
    //初始化图片类
    $album = new Album(-1, $mysql);
    $albumSum = $album->getCatalogAlbumsNum("createtime", $catalogid,
    $mysql);
    $pageInfo = divpage toadmin($albumSum, $currpage, 10);
    if ($currpage <= 1)
    {
        $divpage = "<br />";
    }
    else
    {
$divpage = "<br />
<a href=\"list.php?sid=".$sid."&amp;catalogid=".$catalogid."&amp;page=
".($currpage-1)."\">".gb2utf8("看看前面的")."</a> ";
    }
    if ($currpage >= $pageInfo["max page"])         //判断页面跨度
    {
$divpage .= "<br />\n";
    }
    else
    {
$divpage="<a href=\"list.php?sid=".$sid."&amp;catalogid=".$catalogid.
"&amp;page=".($currpage+1)."\">".gb2utf8("更多相册...")."</a><br />\n";
    }
```

```
    $albumsInfo = $album->searchByShowType("createtime", $catalogid,
    $pageInfo["currpage"], 10, 'desc', $mysql);
    mysql close($mysql);                              //关闭数据库连接
    output waphead();
?>
<wml>
<card title="album list">
    <p>
<?echo gb2utf8($catalog[$catalogid]['name']);?>    //输出分类栏目名称
<br />
<?
foreach ($albumsInfo as $row)                       //遍历栏目变量输出内容
{
echo "<img src=\"point.gif\" alt=\".\" />
<a href=\"album.php?sid=".$sid."&amp;aid=".$row['aid']."\">";
    echo gb2utf8($row['title']);
    echo "</a><br />\n";
}
echo $divpage;  //翻页链接
?>
<br />
<?echo "===".gb2utf8("频道导航")."===";?>
<br />
<a href="list.php?sid=<?=$sid?>&amp;catalogid=1"><?echo gb2utf8($catalog
[1]['name']);?></a> |
<a href="list.php?sid=<?=$sid?>&amp;catalogid=2">
<?echo gb2utf8($catalog[2]['name']);?></a><br />
<a href="list.php?sid=<?=$sid?>&amp;catalogid=3">
<?echo gb2utf8($catalog[3]['name']);?></a> |
<a href="list.php?sid=<?=$sid?>&amp;catalogid=4">
<?echo gb2utf8($catalog[4]['name']);?></a><br />
<a href="list.php?sid=<?=$sid?>&amp;catalogid=5">
<?echo gb2utf8($catalog[5]['name']);?></a> |
<a href="list.php?sid=<?=$sid?>&amp;catalogid=6">
<?echo gb2utf8($catalog[6]['name']);?></a><br />
<a href="list.php?sid=<?=$sid?>&amp;catalogid=7">
<?echo gb2utf8($catalog[7]['name']);?></a><br />
<br />
<br />
<a href="/photo/?sid=<?=$sid?>">
<?echo gb2utf8("返回首页");?></a>
</p>
</card>
</wml>
```

在专辑图片页面点击“返回列表页”，或者通过 Opera 浏览器中加载 list.php 文件执行下面的链接：

```
http://localhost/wap/photo/list.php?sid=f6b4-
8263bb217d2cec1d72ceb84ae6e5&catalogid=0
```

专辑图片的效果如图 14.13 所示。

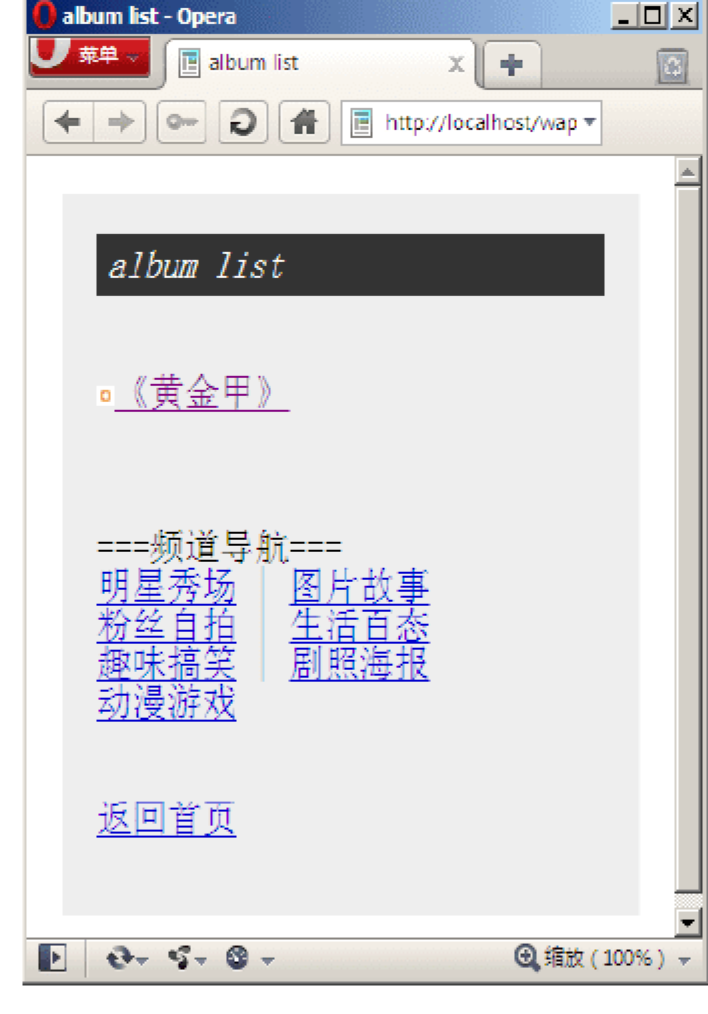


图 14.13　专辑列表

14.5.4　图片显示程序

图片显示程序（showpic.php）的主要功能是显示用户选择的专辑中的图片，并提供给用户下载链接（大图、中

图、小图）。下面请看详细的代码实现。创建文件名为 showpic.php 的 PHP 脚本文件，代码如下：

```
<?
/*******************************************
'文件名：showpic.php
'主要功能：图吧专辑图片显示程序
'说明：
'*******************************************/
    include("include.mysql.php");              //加载 MySQL 连接配置
    include("include.function.php");           //加载函数库
    include("class.pic.php");                  //加载图片处理类
    $sid = init_sid();                         //获得 sid
    $pid = var_process("get", "pid", 177511);  //获得图片 id
    $aid = var_process("get", "aid", 7181);
    $pic = new Pic($pid, $mysql);              //实例化图片类
    mysql_close($mysql);
    output_waphead();
?>
<wml>
<card title="large pic">
<p>
<img src="./<?=$pic->picPath ?>/<?=$pic->picName?>" alt="." /><br />
<?php
echo "<a href=\"/wap/bill.php?sid=".$sid."&amp;downloadurl=".urlencode
("http://localhost/wap/photo/d.php?pid=".$pid."&amp;size=1")."&amp;
backurl=".urlencode("http://localhost/wap/photo/album.php?aid=".$aid.
"&amp;sid=".$sid)."\">".gb2utf8("下载（大）")."</a><br />\n";
echo "<a href=\"/wap/bill.php?sid=".$sid."&amp;downloadurl=".urlencode
("http://localhost/wap/photo/d.php?pid=".$pid."&amp;size=2")."&amp;
backurl=".urlencode("http://localhost/wap/photo/album.php?aid=".$aid.
"&amp;sid=".$sid)."\">".gb2utf8("下载（中）")."</a><br />\n";
echo "<a href=\"/wap/bill.php?sid=".$sid."&amp;downloadurl=".urlencode
("http://localhost/wap/photo/d.php?pid=".$pid."&amp;size=3")."&amp;
backurl=".urlencode("http://localhost/wap/photo/album.php?aid=".$aid.
"&amp;sid=".$sid)."\">".gb2utf8("下载（小）")."</a><br />\n";
?>
<a href="album.php?aid=<?=$aid?>&amp;sid=<?=$sid?>">
<?echo gb2utf8("返回相册");?></a><br />
<a href="/photo/?sid=<?=$sid?>"><?echo gb2utf8("返回首页");?></a>
</p>
</card>
</wml>
```

【代码解读】

上面的代码中，加载图片处理类文件 include("class.pic.php");并初始化，应用其中的方法获得图片路径和名称的信息。

注意： 在输出 WML 链接时，使用 PHP 内置的 urlencode()函数来处理输出的链接，这样可以避免程序转译时出错。读者在处理输出链接时可以使用该函数。

在专辑图片页面选择图片点击“大图”链接，或者通过 Opera 浏览器中加载 list.php 文件执行下面的链接：

```
http://localhost/wap/photo/showpic.php?sid=f6b48263bb217d2cec1d72ceb84a-
```

```
e6e5&aid=22887&pid=408241
```

专辑图片显示的效果如图 14.14 所示。



图 14.14　专辑图片显示

14.5.5　图片下载程序

图像下载程序（download.php）的主要功能是根据用户选择的图片（大图、中图、小图）给出下载链接，用户点击该链接后就可以将图片下载到手机上。下面请看详细的代码实现。创建文件名为 download.php 的 PHP 脚本文件，代码如下：

```
<?
/*******************************************
'文件名：download.php
'主要功能：图片下载程序
'说明：
 *******************************************/
include ("include.mysql.php");                    //加载 MySQL 连接配置
include ("include.function.php");                 //加载函数库
include ("class.pic.php");                        //加载图片处理类
$pid = var_process ( "get", "pid", 177511 );      //获得图片 id
$size = var process ( "get", "size", 1 );  //1 = large, 2=middle, 3=small
$pic = new Pic ( $pid, $mysql );                  //初始化图像类
//根据尺寸信息设定路径参数
if ($size == 1)
    $downloadurl = "http://localhost/wap/photo/" . $pic->picPath . "/240 " .
    $pic->picName;
else if ($size == 2)
    $downloadurl = "http://localhost/wap/photo/" . $pic->picPath . "/128 " .
    $pic->picName;
else
    $downloadurl = "http://localhost/wap/photo/" . $pic->picPath . "/80_" .
    $pic->picName;
mysql close ( $mysql );
output_waphead ();
?>
<wml>
<card title="download">
<?
if ($downloadurl != "") {
    echo "<p>" . gb2utf8 ( "请点击下载您缓存的内容" ) . "<br /> //输出提示内容
<a href=\"" . $downloadurl . "\">" . gb2utf8 ( "点击下载" ) . "</a></p>\n";
}
?>
<p><a href="http://localhost/wap/index.php">   //输出导航链接
<?
echo gb2utf8 ( "返回 WAP 首页" );                  //输出导航提示
?></a>
</p>
</card>
</wml>
```

选择一个尺寸的图片点击“下载”，进入下载页面，显示的效果如图 14.15 所示。查看

页面源代码可以发现，“点击下载”链接已经给出了图片的真实地址，单击该链接就可以将图片下载到用户手机上。



图 14.15　图片下载程序

14.6 小　　结

WAP 技术通过将无线移动通信技术和 Internet 结合起来，通过通用的平台将目前 Internet 中广泛使用的 HTML（Hypertext Markup Language）语言描述的信息转换成用 WML（Wireless Markup Language）语言描述的信息，并在移动设备上显示。在这个过程中，结合一定的内容载体就构成了一个典型的 WAP 应用。

WAP 资讯模块是一个典型的 PHP 与 WAP 结合的应用。第 14.1 节从 WAP 应用基础讲起，包括 WAP 系统概述、WML 基础、WAP 浏览器安装与调试。第 14.2 节开始数据库的设计，按照需求分为 3 个表，即用户资料表、专辑资料表、图片资料库表。在数据库的实际应用上也是对模块组成和功能的梳理。第 14.3 节对创建模块用到的核心函数进行详细介绍。第 14.4 节介绍 WAP 认证程序。由于 WAP 应用的特殊性，需要对内容及用户身份做必要的认证后才能得到相应的服务，本章正是完成这样一个功能。在做好了全部的准备工作后，第 14.5 节开始了业务部分的程序开发，这部分需要实现业务包括 WAP 资讯首页显示、专辑图片页面显示、专辑列表页面显示、图片显示程序和图片下载程序。以上部分共同构成了一个典型的、完整的使用 PHP 构建的 WAP 应用，读者在学习的时候要注意对业务流程的梳理和理解。

第 15 章　广告管理模块（JavaScript+iframe+Smarty）

广告管理模块是 PHP 生成并管理 JavaScript 文件的典型应用。通过对本章的学习，读者可以对 PHP 生成和管理文件有清晰的了解，在代码实现方面可以对具体实现文件生成和管理的有一个直观的认识。

本章主要涉及的知识点如下。

- preg_replace()函数：PHP 的内置函数功能是执行正则表达式的搜索和替换。
- str_replace()函数：PHP 的内置函数功能是字符串取代，是文件管理的常用函数。
- Javascript:history.go(-1)函数：JavaScript 函数，该函数可以返回浏览器的前一次操作，在文件管理中常被用到。
- 程序流程控制实现方法：在业务类的应用中，通过判断动作类型（action）来实现不同业务的区分，从而实现程序的流程控制。

15.1　广告管理概述

广告管理系统，又被称为网络广告管理系统、互联网广告管理系统、在线媒体广告管理系统。该系统是随着互联网广告市场的发展应运而生的，目的是帮助网站主和广告主轻松实现专业的广告投放、管理和数据统计等工作的管理系统，在技术层面它的主要功能是完成对 JavaScript 脚本的生产与管理。

15.1.1　广告管理基本特征

随着网络广告市场的发展，网络媒体策划和产品营销人员需要更加细致地管理，及时进行广告优化，确保网络广告资源被高效使用。同时，网络管理者需要更加灵活地组织和调配网络资源，在确保精确的广告投递的前提下，依托广告管理系统的技术基础，与销售团队进行深层次的整合，形成多样性的销售方案。因此，广告管理系统的产生就是为了满足各种市场推广方面的需求。

广告管理的形成经历了一个演变的阶段，简单来说可以分为 3 个阶段：

（1）最初的直接手工修改页面代码阶段。

（2）通过管理 JavaScript 或 iframe 的方式来管理广告阶段。

（3）后来部分内容管理系统（CMS）里面引入了广告管理模块，通过插件的方式来实

现广告的管理阶段。

由于广告投放方式日趋复杂，要求和分析方式越来越细化，上面提到的几种方式已经很难适应广告的管理。随后出现了专门的广告管理系统，负责广告投放和管理并完成辅助的统计和用户管理功能。广告管理系统的基本特征如下。

- 广告生成：根据设定的规则生成指定的广告条目。
- 广告投放：投放系统生成的广告条（目录和命名规则管理是核心）。
- 广告管理：对选定条目的广告进行编辑、添加、修改、删除等操作。
- 多类型支持：支持多种广告类型，比如 HTML 格式、JavaScript 脚本、Flash 格式、和其他格式的广告。

以上是广告管理系统的核心功能和基本的特征。下面列出了广告管理的辅助功能，是广告管理系统常见的扩展应用。

- 广告分组：以组为单位对广告分类，可以更有效地进行管理。
- 投放计划：对同组内的广告，可以根据投放计划需要设置不同显示比。设置广告显示或点击达到一定次数后停止播放。
- 统计功能：包括显示、点击和点击率的时、日、月统计。
- 搜索功能：使查看报表更加快捷方便。
- 统计报表：报表包括今日简报，以某个广告组或广告条为对象的详细分析，以及多个广告组或广告条之间的对比分析。
- 用户管理功能：细分的用户权限，使系统管理更加系统化，满足大型网站复杂管理的需要。

15.1.2　工作流程描述

广告管理模块的程序按照功能分为前台显示和后台管理两个部分，具体的程序工作流程如图 15.1 所示。

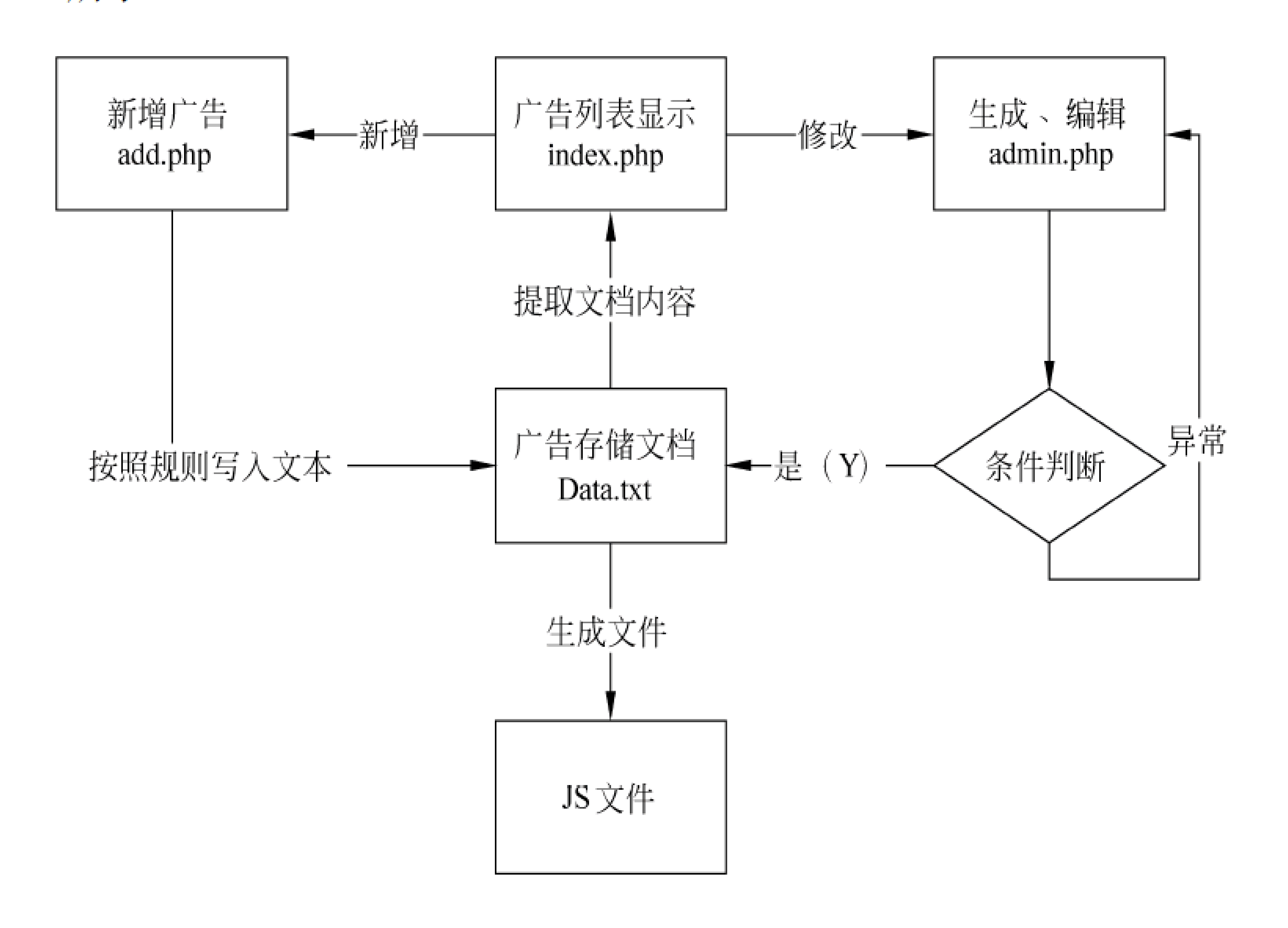


图 15.1　广告管理模块工作流程图

15.2　文件目录结构设计及说明

通过对广告管理系统工作流程的介绍，结合前文提到的广告管理系统的基本特征和核心功能，本节将规划广告系统的文件目录，并说明各个文件需要完成的功能。文件目录和文件说明相当于一个基础的框架，按照说明的要求逐步补齐文档中的内容，最终就能实现预先的设计目标。请读者在学习本章时，仔细体会设计的细节，为以后独立进行项目分析和任务拆分打下良好的基础。

15.2.1　系统根目录说明

首先建立文件名为 ad 的文件夹，将 ad 文件夹作为广告管理的根目录，所有的广告管理相关的文件都存放在该子目录中。

然后，在 ad 文件夹下创建 8 个文件夹和 4 个 PHP 脚本文件，目录结构如图 15.2 所示。下面分别介绍每个文件具体的功能。



图 15.2　广告管理模块文件目录

- index.php：系统入口文件，显示广告管理界面和操作面板。
- admin.php：广告条目修改程序，显示广告修改该界面，完成对应广告条目的广告内容修改。
- add.php：新增广告条目程序，显示新增广告界面，完成添加新增广告功能。
- del.php：删除广告条目程序，显示删除广告提示界面，完成指定广告条删除操作。
- config 文件夹：配置文件文件夹。
- inc 文件夹：系统库文件文件夹。
- js 文件夹：生成 JavaScript 广告文件夹。

- Smarty 文件夹：模板引擎文件夹。
- tpl 文件夹：广告管理模板文件夹。
- image 文件夹：图片文件夹。
- css 文件夹：CSS 样式表文件夹。
- bak 文件夹：备份文件文件夹。

15.2.2　配置文件目录说明

配置文件目录是存放广告系统配置文件的文件夹，该文件夹一共包含 4 个文件目录，结构如图 15.3 所示。下面具体说明这 4 个文件的功能。

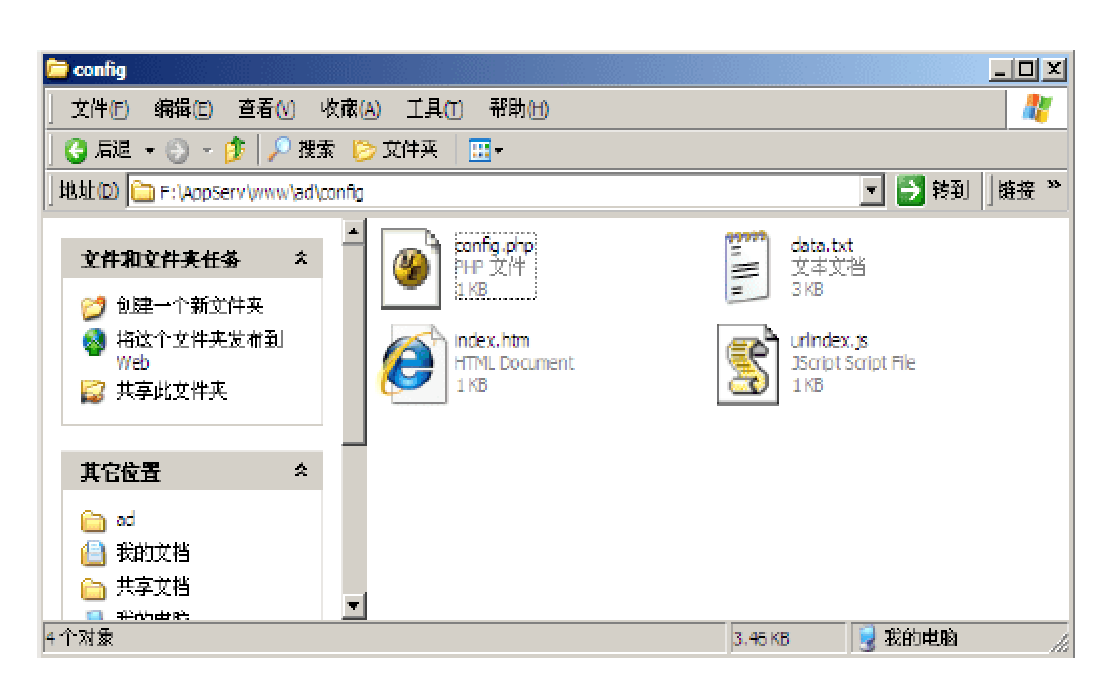


图 15.3　配置文件目录

- config.php 文件：预定义系统变量，包括管理员账号信息、广告代码存放的目录等。
- data.txt 文件：广告代码存放文件。
- index.htm 文件：管理框架文件。
- urlindex.js 文件：管理文件加载 JavaScript 文件。

15.2.3　函数库文件目录

函数库文件目录是存放公共函数库的文件夹，该文件夹一共包含两个文件。目录结构如图 15.4 所示，下面具体说明这两个文件的功能。

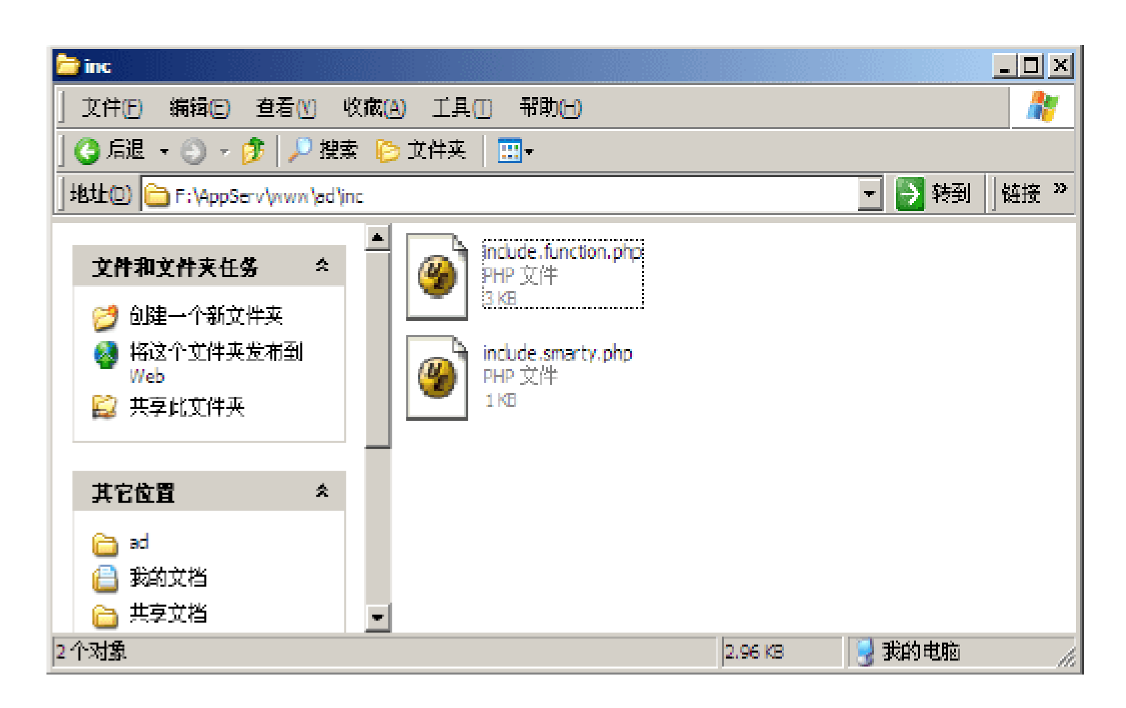


图 15.4　函数库文件目录

- include.function.php 文件：广告管理系统公共函数库。
- include.smarty.php 文件：Smarty 模板引擎配置文件。

15.2.4　模板引擎目录

模板引擎目录是存放整个模板引擎模块文件（使用的是开源第三方模板引擎 Smarty），为了方便读者理解模板引擎的运行过程，对模板引擎目录做简要的说明，方便读者参考。该文件目录中共包含 6 个文件文件夹和 4 个文件，目录结构如图 15.5 所示。下面对核心的文件做简要的说明。

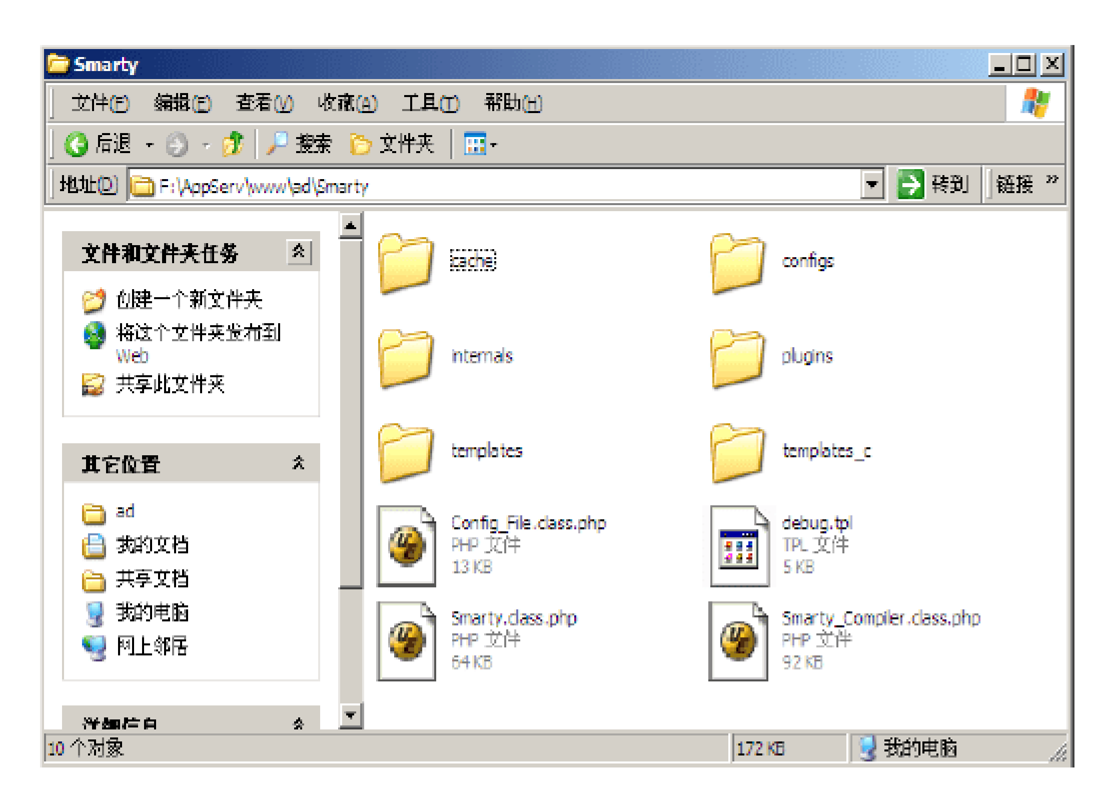


图 15.5　Smarty 模板引擎文件目录

- ❑ Smarty.class.php：模板引擎类文件。
- ❑ Config_File.class.php：配置文件类文件。
- ❑ Smarty_Compiler.class.php：模板编译类文件。
- ❑ Cache 文件夹：文件缓存文件夹。
- ❑ Configs 配置文件夹：配置文件文件夹。
- ❑ Plugins 文件夹：插件文件。
- ❑ Templates 文件夹：系统默认模板文件夹。
- ❑ debug.tpl 文件：异常抛出页面模板。

15.2.5　广告管理模板目录

广告管理模板目录是存放广告管理模板文件夹，该文件夹一共包含 4 个文件，目录结构如图 15.6 所示。下面具体说明这 4 个文件。

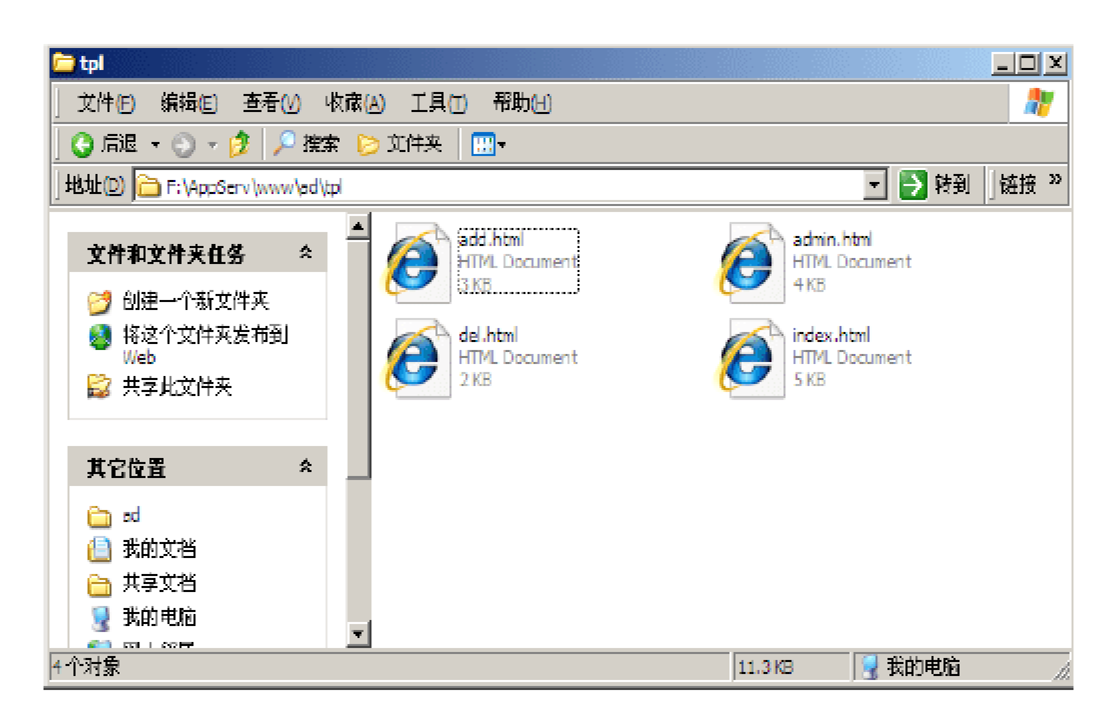


图 15.6　广告管理模板文件目录

- index.html：广告管理主页面模板。
- admin.html：广告编辑页面模板。
- add.html：新增广告页面模板。
- del.html：删除广告页面模板。

15.3　核心函数说明

广告管理模块的开发中，很多基础的功能复用率很高，因此将这些核心的功能封装成公共函数库（include.function.php），在完成具体的程序时，就不用过多的考虑程序功能方面的实现，只需要关注业务逻辑和流程控制。以下将详细介绍广告管理模块中的核心函数并做相应的说明。

15.3.1　使用 allen_thu()函数实现搜索和替换

该函数的功能是执行正则表达式的搜索和替换。函数代码如下：

```
// 执行正则表达式的搜索和替换
function allen thu($strc){
$strc=str_replace("'","\'",$strc);              //字符替换
$strc=preg_replace("~(?:\r)?\n~s"," ",$strc);  //正则规则匹配
$strc="document.write('$strc');";               //打印输出
  return $strc;
}
```

【代码解读】

使用 PHP 内置的 str_replace()函数，对字符串替换处理后的字符串，再用正则匹配函数 preg_replace()匹配结果，然后用 document.write()输出。

注意：document.write(');函数是 JavaScript 中的函数，需要在浏览器中输出打印值。

15.3.2　使用 allen_file()函数生成文件函数

该函数的功能是生成文件函数，具体地说，就是生成广告的.js 文件。函数代码如下：

```
//生成文件函数
function allen file($data,$file){
    $fp=fopen($file,"w+");                      //创建文件操作的句柄
    $data=StripSlashes("$data");                //去掉反斜线字符
    fwrite($fp,"$data");
    if (!is_file($file)){
        die("文件$file 写入失败，可能是您写入该文件所在的文件夹不存在，请检查!");
    }
    fclose($fp);                                //关闭句柄
}
```

【代码解读】

创建一个文件操作的句柄，使用 StripSlashes()函数去掉反斜线字符，然后使用 fwrite()函数将内容写入文件中，最后关闭文件句柄。

15.3.3　使用 isok()函数实现页面跳转

该函数的功能是在操作成功后显示跳转的页面，该页面会有提示操作的面板。函数代码如下：

```
//成功跳转页面
function isok($isok){
$output="<html>
<head>
<head>
<link rel=\"stylesheet\" href=\"css/style.css\">
<table cellspacing=1 cellpadding=3 width=55% bgcolor=\"#000000\" class=
\"fonts\" align=\"center\">
  <tr bgcolor=\"#9999ff\">
    <td height=\"11\" align=\"center\" bgcolor=\"#999999\" class=\
    "mycss\">
      <div align=\"left\"> 系统消息:成功!</div>
    </td>
  </tr>
  <tr>
    <td align=\"center\" bgcolor=\"#FFFFFF\" class=\"mycss\">
        <table width=\"100%\" border=\"0\" cellspacing=\"0\" cellpadding=
        \"3\">
          <tr>
            <br>
          <td colspan=\"2\"><font color=\"red\">".$isok."</font>
<p><a href=\"index.php\">返回管理首页
              </a></p><p><a href=\"add.php\">添加新的广告代码
              </a></p>
          </td>
          </tr></table>
</table>";
echo $output;
exit;
}
```

【代码解读】

通过 echo()函数输出编辑好的 HTML 代码，内容为提示操作成功，并引导用户下一步的操作。

15.3.4　使用 error()函数报错

该函数的功能是在操作失败或者是程序异常时，抛出报错页面并给出报错原因。

具体的操作是通过 echo()函数输出编辑好的 HTML 代码，内容为提示操作失败，并给出程序异常的原因。

函数代码如下：

```
//异常错误页面
```

```
function error($error){
$output="<link rel=\"stylesheet\" href=\"css/style.css\">
<table cellspacing=1 cellpadding=3 width=55% bgcolor=\"#000000\" class=
\"fonts\" align=\"center\">
  <tr bgcolor=\"#9999ff\">
    <td height=\"11\" align=\"center\" bgcolor=\"#999999\" class=\"mycss\">
      <div align=\"left\"> Error 错误原因可能是:</div>
    </td>
  </tr>
  <tr>
    <td align=\"center\" bgcolor=\"#FFFFFF\" class=\"mycss\">
        <table width=\"100%\" border=\"0\" cellspacing=\"0\" cellpadding=
        \"3\">
        <tr>
            <br>
        <td colspan=\"2\"><font color=\"red\">".$error."</font>
            <p><a href=\"javascript:history.go(-1);\">&lt;&gt; 点击这里返回
            上一页 &lt;&gt;
              </a></p>
          </td>
          </tr></table>
</table>";
exit;
}
```

【代码解读】

以上两个函数都是通过 PHP 输出 HTML。这里需要注意的是，HTML 代码涉及标签值需要用“\”符号隔离，否则系统会报错。这里还有个小技巧，返回上一步骤的操作可以使用 javascript:history.go(-1)函数，示例如下：

```
<p><a href=\"javascript:history.go(-1);\">&lt;&gt; 点击这里返回上一页 &lt;
&gt; </a></p>
```

将会返回浏览器前一步操作的页面，读者在今后的编程中可以尝试使用，可以为用户提供更好的使用体验。

15.4　广告管理界面程序

广告管理程序的功能是用户通过操作界面的功能，引导对广告投放（维护）进行日常的管理。广告管理模块的程序可以分为两个部分：

- 管理界面程序。它主要实现的功能是内容搜索、生成管理面板等。本节将介绍这部分的内容。
- 广告管理的操作部分。它包括广告生成、新增广告操作、广告编辑操作、广告删除操作等内容，将在 15.5 节具体介绍。

15.4.1　显示广告列表

广告管理程序的核心是广告页面展示的部分，它的作用相对一个引导页面所有的操作，都是从看到广告列表的显示开始发起的。首先来看页面列表显示部分的程序实现，代

码如下：

```
/*******************************************
'文件名：index.php
'主要功能：广告管理模块 index
'说明：
'*******************************************/
include_once(dirname(__FILE__)."/inc/include.smarty.php");
                                                //加载 Smarty 配置文件
include(dirname(__FILE__). "/inc/include.function.php");  //加载函数库文件
require(dirname(__FILE__)."/config/config.php");          //加载容配置文件
$type=$_REQUEST[type];                                    //获得操作类型
$kw=$_REQUEST[kw];
$page=$_REQUEST[page];                                    //获得当前页数
if($page<1){
    $page=1;
}
$page=$page-1;
$msgd=explode("|www_allen|",file_get_contents($dbtable)); //切割字符串
$rows=count($msgd);                                       //计算行数
if($kw){
for ($i=0;$i<=$rows;$i++){
    $tmp2=explode("|allen|",$msgd[$i]);
    if($type=="name"){                                //判断操作类型(name)
        $msgn[$i]=$tmp2[0];
        }elseif($type=="nr"){
            $msgn[$i]=$tmp2[2];
        }
        elseif($type=="gge"){                         //类型判断（gge）
            $msgn[$i]=$tmp2[4];
        }
    else{$msgn[$i]=$tmp2[3];
}
if(eregi($kw,$msgn[$i])){                             //字符串比对解析
    $chj_nr[]=$msgd[$i];
    $chj_numib++;
    }
}
if($chj_numib<1){
    echo "没有相关 $kw 的匹配结果!!";                   //输出报错信息
    exit;
}
$msg=array_reverse($chj_nr);
    $total=ceil($chj_numib/$pagesize);                //计算总页面数
    $kwnum=-1;
    }else{
        $chj_numib=$rows;
        $msg=array_reverse($msgd);
    $total=ceil($chj_numib/$pagesize);                //计算分页
    }
//如果每页显示数*总页数小于留言总条数则
if($pagesize*$total< $chj_numib){
    $total++;                                         //总页面数加 1
}
$total2=$total-1;
$page2=$page+1;
$pp=$page*$pagesize+1+$kwnum;                         //计算开始条数
```

```
$pp2=$pp+$pagesize;                                    //计算结尾条数
$nextpage=$page+2;                                     //下一个页面
$prevpage=$page;                                       //上一个页面
$chj adlen=$chj numib-1;
for($i=$pp;$i< $pp2;$i++){
if($i < $chj numib){
    $tmp=explode("|allen|",$msg[$i]);
    if(!empty($kw)){
        //有搜索值时
        $arr[$i]['title']=$tmp[0];
        $arr[$i]['type']=$tmp[1];
        $arr[$i]['size']=$tmp[4];
        $arr[$i]['desc']=$tmp[3];
        }else{
        //没有搜索时即显示全部
        $arr[$i-1]['title']=$tmp[0];
        $arr[$i-1]['type']=$tmp[1];
        $arr[$i-1]['size']=$tmp[4];
        $arr[$i-1]['desc']=$tmp[3];
    }
    }
}
    $smarty->assign("kw", $kw);
    $smarty->assign("type", $type);                    //类型标签
    $smarty->assign("prevpage", $prevpage);            //前一页标签
    $smarty->assign("nextpage", $nextpage);            //下一页标签
    $smarty->assign("total2", $total2);                //页面标签
    //总条数标签
    $smarty->assign("page2", "<font color=red>".$page2."</font>/".
    $total."");
    $smarty->assign("total", $chj_adlen);              //总条数标签
    $smarty->assign("content", $arr);                  //生成广告内容标签
    $smarty->display("./index.html");                  //加载首页模板
```

代码的执行流程步骤如下。

（1）加载配置文件（模板引擎配置文件、公共函数库、预定义配置参数）。

（2）判断操作的类型（是否为关键字搜索），使用$kw 变量做流程控制。

（3）获取广告内容文件，使用 explode()对预先定义的分割标签切割字符串，该条代码如下：

```
$msgd=explode("|www_allen|",file_get_contents($dbtable));
```

（4）在通过循环遍历，将数组中的字符串取出处理后的再次切割为“内容”，处理过程代码如下：

```
for($i=$pp;$i< $pp2;$i++){
if($i < $chj numib){
    $tmp=explode("|allen|",$msg[$i]);  //切割字符
    if(!empty($kw)){                   //判断是否为空
        $arr[$i]['title']=$tmp[0];
        $arr[$i]['type']=$tmp[1];
        $arr[$i]['size']=$tmp[4];
        $arr[$i]['desc']=$tmp[3];
        }else{
        //当为空时为前一个节点数组赋值
        $arr[$i-1]['title']=$tmp[0];
```

```
        $arr[$i-1]['type']=$tmp[1];
        $arr[$i-1]['size']=$tmp[4];
        $arr[$i-1]['desc']=$tmp[3];
    }
    }
}
```

（5）得到结果数组之后生成 Smarty 模板标签，代码如下：

```
$smarty->assign("kw", $kw);
$smarty->assign("type", $type);
$smarty->assign("prevpage", $prevpage);
$smarty->assign("nextpage", $nextpage);
$smarty->assign("total2", $total2);        //页面标签
//总条数标签
$smarty->assign("page2", "<font color=red>".$page2."</font>/".$total."");
$smarty->assign("total", $chj adlen);      //总条数标签
$smarty->assign("content", $arr);          //生成广告内容标签
$smarty->display("./index.html");          //加载首页模板
```

当得到模板的数组标签后，需要加载广告管理主页的模板（index.html），将显示内容部分替换成对应的 Smarty 标签，输出给显示器就完成了列表的显示。核心的页面代码如下：

```
{|section name=content loop=$content|}
<table cellSpacing="1" cellPadding="3" width="760" align="center"
    bgColor="#999999" border="0">
    <tbody>
        <tr bgColor="#c0c0c0">
            <td width="132" height="25">名称：{|$content[content].title|}
            </td>
            <td width="126">广告类型：{|$content[content].type|}</td>
            <td width="101">规格：{|$content[content].size|}</td>
            <td width="239">管理： <a href="admin.php?action=edit&id=
            {|$content[content].title|}">编辑</a>  
            <a onClick="return delad();"href="del.php?action= del&id=
            {|$content[content].title|}">删除广告</a></td>
            <td width="126"><a href="#" target="_self">&gt;&gt;回顶端
            &lt;&lt;</a>
            <input type="checkbox" name="idjs[]"value="{|$content [content].
            title|}"> 生成</td>
        </tr>
        <tr bgColor="#e6e6e6">
            <td width="132" height="24">代码说明：</td>
            <td colSpan="4" height="24">{|$content[content].desc|}</td>
        </tr>
        <tr bgColor="#e6e6e6">
            <td width="132" height="23">JS 广告调用代码：</td>
            <td colSpan="4" height="23"><input name="Input"
                onFocus="this.select()"
                value='&lt;script type="text/javascript" src="/js/{
                |$content[content].title|}.js" charset="gb2312"&gt;
                &lt;/script&gt;'
                size="58" /> <input name="Input2" onFocus="this.select()"
                value='&lt;script type=\"text/javascript\" src=\&quot; /js
                /{|$content[content].title|}.js\&quot;&gt;&lt;/script&gt;'
                size="20" /></td>
        </tr>
```

```
        <tr>
            <td colSpan="5" height="8">代码预览： <a
                href="./config/index.htm?js/{|$content[content].title|}"
                target="adm">新窗口中预览</a> <iframe frameborder=0 width=
                100% height=100%
                src="./config/index.htm?js/{|$content[content]. title|}">
                </iframe>
            <div
                style="OVERFLOW-Y: scroll; OVERFLOW-X: hidden; WIDTH: 100%;
                HEIGHT: 100px; BACKGROUND-COLOR: #ffffff">
            </div>
            </td>
        </tr>
    </tbody>
</table>
{|/section|}
```

【代码解读】

在 Smaty 的模板标签规则中处理遍历数组是最为常见的应用，它被广泛地应用在列表显示一类的程序中，它的标签格式如下：

```
{|section name=content loop=$content|}
中间部分为循环体
{|/section|}
```

注意：在循环标签的所有内容都将被循环，包括 HTML 标签，这个特性使得页面模板的代码量极大地减少，逻辑更为清晰。读者在使用模板标签的时候可以仔细体会它带来的优势。

在浏览器中运行下面的链接，可以看到完整的显示效果，如图 15.7 所示。

```
http://localhost/ad/index.php
```



图 15.7　广告管理列表页

15.4.2　内容搜索的实现

内容搜索是在广告列表显示内容的基础上做关键字的匹配，然后重新生成数组。因为与广告列表共用一个页面，所以页面代码和标签都是完全相同的，这里不再重复了，请读者参考前面的页面代码。下面具体地介绍内容搜索的代码实现。

（1）页面搜索的表单代码：

```
<form action="" method="get" name="form1" id="form1">
<tr>
    <td align="right"><input name="kw" type="text" id="kw"
        value="{|$kw|}"> <select name="type" id="type">
        <option value="nr">广告内容</option>
        <option value="name">广告 ID</option>
        <option value="all">广告备注</option>
        <option value="gge">广告规格</option>
    </select> <input type="submit" value="搜索"> <a href="add.php"
        target="_self">添加新的广告</a></td>
</tr>
</form>
```

【代码解读】

上面的代码定义了一个标准的表单，提供给用户 4 种类型的选项。其中，input 标签的值就是用户输入的查询关键字，代码如下：

```
<input name="kw" type="text" id="kw" value="{|$kw|}">
```

（2）接收到了表单传过来的查询关键字，PHP 脚本匹配关键字，生成结果集数组，然后转化成对应的模板标签并输出查询结果。下面来看 PHP 是如何处理关键字获得查询结果的。先加载配置文件并获得查询类型，代码如下：

```
include_once(dirname(__FILE__)."/inc/include.smarty.php");
                                                //加载 Smarty 配置文件
include(dirname(__FILE__). "/inc/include.function.php"); //加载函数库文件
require(dirname(__FILE__)."/config/config.php");        //加载容配置文件
$type=$_REQUEST[type];                                  //获得操作类型
$kw=$_REQUEST[kw];
$page=$_REQUEST[page];
```

【代码解读】

上面的代码将获得的关键字赋值给$kw 变量，这里需要具体解释配置文件 config.php 的作用，请看 config.php 文件的内容：

```
/*******************************************
'文件名：config.php
'主要功能：配置文件
'说明：
'*******************************************/
//定义管理员信息
$adminname="admin"; //管理员账号
$adminword="admin"; //管理员密码
//管理员 URL 安全码
```

```
//如设置后，管理地址将是 http://你的网址/本程序目录/log.php?safecode
$safecode="safecode";
//设置配置文件信息
//广告代码存放的目录
$addata="js";
$dbtable="config/data.txt";//设置文件存放位置
$pagesize="10";              //每页显示的广告条数
```

【代码解读】

以上代码定义了管理员信息的变量和广告代码存放的目录。可以看到，广告代码文件是 data.txt 文件，也就是说，查询关键字是与这个文档中的内容做匹配得到的查询结果。读者要注意，这里的逻辑关系就是结果集的来源。

（3）在直观地了解了结果集来源后，来看程序是如何读取和匹配结果的。通过 file_get_contents()函数获得广告文件中的内容，再通过 explode()函数切割成字符串数组，代码如下：

```
$msgd=explode("|www_allen|",file_get_contents($dbtable));
```

（4）匹配规则和处理数组代码如下：

```
if($kw){
for ($i=0;$i<=$rows;$i++){
    $tmp2=explode("|allen|",$msgd[$i]);     //切割字符
    if($type=="name"){                      //类型赋值
        $msgn[$i]=$tmp2[0];                 //消息内容
        }elseif($type=="nr"){
            $msgn[$i]=$tmp2[2];
        }
        elseif($type=="gge"){
            $msgn[$i]=$tmp2[4];
        }
    else{$msgn[$i]=$tmp2[3];
}
if(eregi($kw,$msgn[$i])){                   //字符串比对解析(与大小写无关)
    $chj_nr[]=$msgd[$i];
    $chj_numib++;
    }
}
if($chj_numib<1){
    echo "没有相关 $kw 的匹配结果!!";
    exit;
}
//处理分页和计算总页数代码省略
...
$chj_adlen=$chj_numib-1;
for($i=$pp;$i< $pp2;$i++){
if($i < $chj_numib){
    $tmp=explode("|allen|",$msg[$i]);
    if(!empty($kw)){
        $arr[$i]['title']=$tmp[0];
        $arr[$i]['type']=$tmp[1];
        $arr[$i]['size']=$tmp[4];
        $arr[$i]['desc']=$tmp[3];
        }else{
        //当为空时为前一个节点数组赋值
        $arr[$i-1]['title']=$tmp[0];
```

```
        $arr[$i-1]['type']=$tmp[1];
        $arr[$i-1]['size']=$tmp[4];
        $arr[$i-1]['desc']=$tmp[3];
    }
    }
}
```

【代码解读】

得到的$arr 数组就是跟关键字匹配得到的结果数组了。现在输入“互换链接”关键字测试程序匹配的结果，得到结果如图 15.8 所示。



图 15.8　搜索结果

15.5　广告管理操作程序

广告管理的操作部分包括广告生成、新增广告操作、广告编辑操作、广告删除操作等内容，这几个管理功能是最常见的，也是最为核心的。以上几个基本功能相互配合就可以有效地管理广告系统。下面分别介绍这些功能是如何实现的。

15.5.1　新增广告操作

新增广告操作的核心是处理新增广告表单中的数据，处理后的数据写入广告数据存放文本（data.txt）中。下面来看页面表单的部分，核心的表单代码如下：

```
<div align="left">
<table width="100%" border="1" cellpadding="3" cellspacing="0"
    bgcolor="#CCCCCC">
    <tr>
        <td width="16%">广告 ID：<font color="#FF0000">*</font><br>
        <br>
        </td>
        <td width="84%"><input name="id" type="TEXT" class="input"
            id="id" size="25" maxlength="30"></td>
    </tr>
    <tr>
```

```
        <td height="14">广告类型<font color="#FF0000">*</font>   
        </td>
        <td height="14"><select name="nx" id="nx">
            <option value="纯 HTML">纯 HTML</option>
            <option value="脚本式">脚本式</option>
        </select> HTML 中的换行符将会被省略掉。</td>
    </tr>
    <tr>
        <td>规格(宽*高)</td>
        <td><input name="gge" type="TEXT" class="input" id="id" value="X"
            size="20" maxlength="30"></td>
    </tr>
    <tr>
        <td height="76" valign="top">
        <p>广告内容<font color="#FF0000">*</font></p>
        </td>
        <td valign="middle"><textarea name="data" cols="80" rows="10"
            wrap="virtual" class="input" id="data" tabindex="2">
            </textarea></td>
    </tr>
    <tr>
        <td height="30">简单注释：</td>
        <td height="30"><textarea name="notice" cols="80" rows="4"
            wrap="virtual" class="input" id="notice" tabindex="2">
            </textarea></td>
    </tr>
    <tr align="center">
        <td height="30" colspan="2"><input type="submit" name="action"
            value="提交" class="botton"> <input type="reset" value="重填"
            class="botton" name="reset"> <input type="button"
            value="返回上一页" onClick="javascript:history.back(-1);"
            class='botton'></td>
    </tr>
</table>
</div>
```

【代码解读】

上述页面模板代码是一个标准的数据表单。其中，表单标签中的值传递给添加广告程序（add.php）处理，添加广告程序将表单传递过来的值，拆分处理成广告存放文件的预定义格式，并写入该文件保存。

添加广告程序（add.php）的代码具体如下：

```
<?php
/*******************************************
'文件名：add.php
'主要功能：添加广告程序
'说明：
'*******************************************/
include_once(dirname(__FILE__)."/inc/include.smarty.php");
                                                //加载 Smarty 配置文件
include(dirname(__FILE__). "/inc/include.function.php");  //加载函数库文件
require(dirname(__FILE__)."/config/config.php");          //加载容配置文件
$rand=rand(0,9999);                                       //生成随机数
if($_REQUEST['action']=="提交"){
$id=$_REQUEST[id];
$nx=$_REQUEST[nx];
```

```
$gge=$ REQUEST[gge];
$notice=$ REQUEST[notice];
$data=$ REQUEST[data];
//处理文件内容
$msgd=explode("|www allen|",file get contents($dbtable));
$rows=count($msgd);
for ($i=0;$i<=$rows;$i++){
$tmp2=explode("|allen|",$msgd[$i]);                   //切割字符串
$msgn[$i]=$tmp2[0];
if(eregi("^$id$",$msgn[$i])){
$chj nr=$msgd[$i];
}
}
if($chj nr!=""){
    error("该广告 ID 已经存在，请换一个吧。");              //报错信息
}
if($data=="" or $id==""){                             //判断数据是否存在
die("ID 或者内容为空！");
}
$fp=fopen("$dbtable","a+");        //打开指定文件'a+'（打开文件方式为可读写）
$all.="$id|allen|$nx|allen|$data|allen|$notice|allen|$gge|www_allen|";
@fputs($fp,$all) or die(error('写入信息时出错！'));
if($nx=="纯 HTML"){
    $data=allen thu($data);
    allen_file($data,"$addata/$id.js");
    }else{
    allen file($data,"$addata/$id.js");
    }
isok("添加广告成功!请稍等...");
}
$smarty->display("./add.html");
?>
```

在浏览器中执行下面的链接测试新增广告：

```
http://localhost/ad/add.php
```

效果如图 15.9 所示。添加测试广告数据后单击“提交”按钮，提示提交成功。提示成功页面如图 15.10 所示。

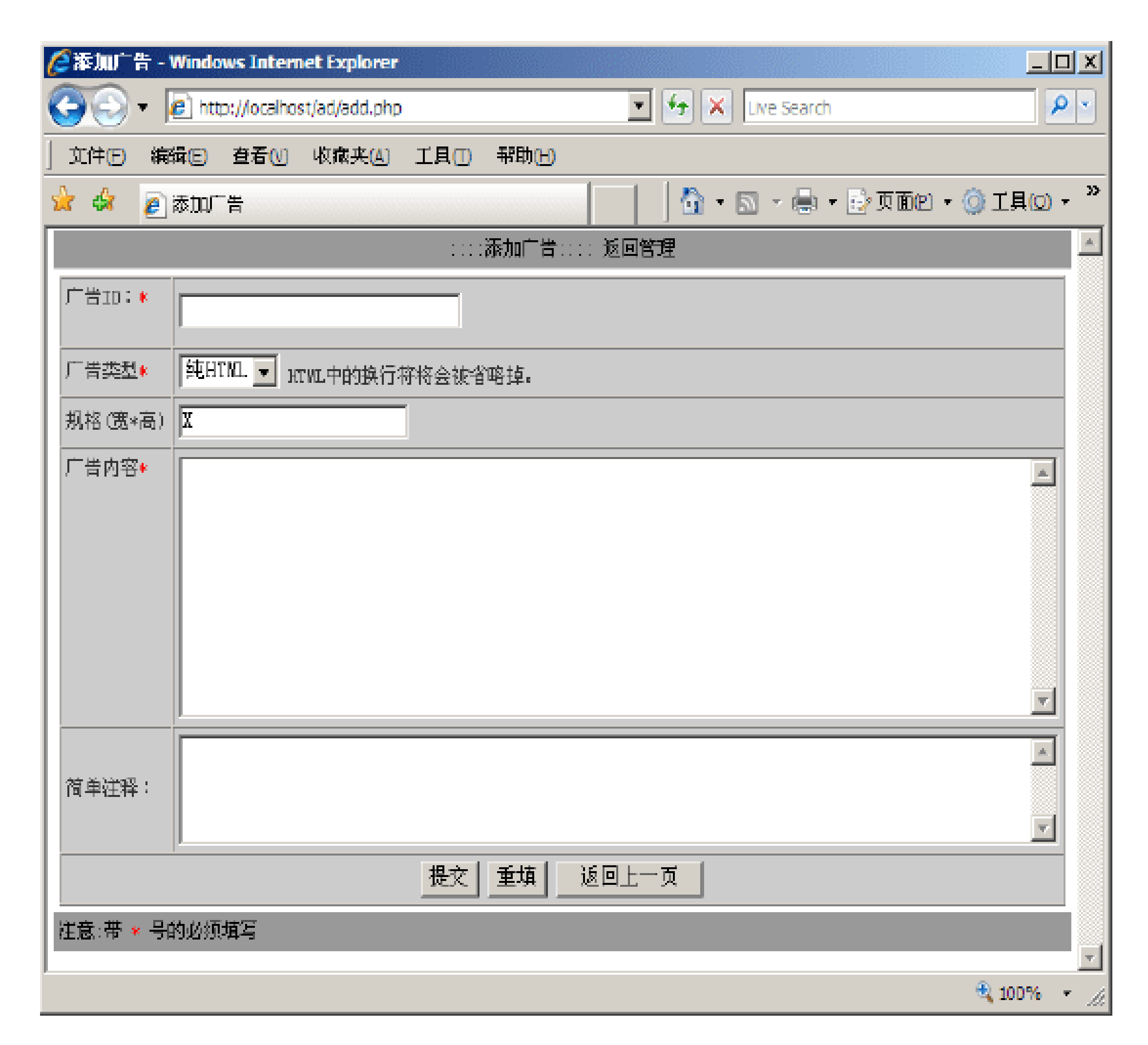


图 15.9　新增广告

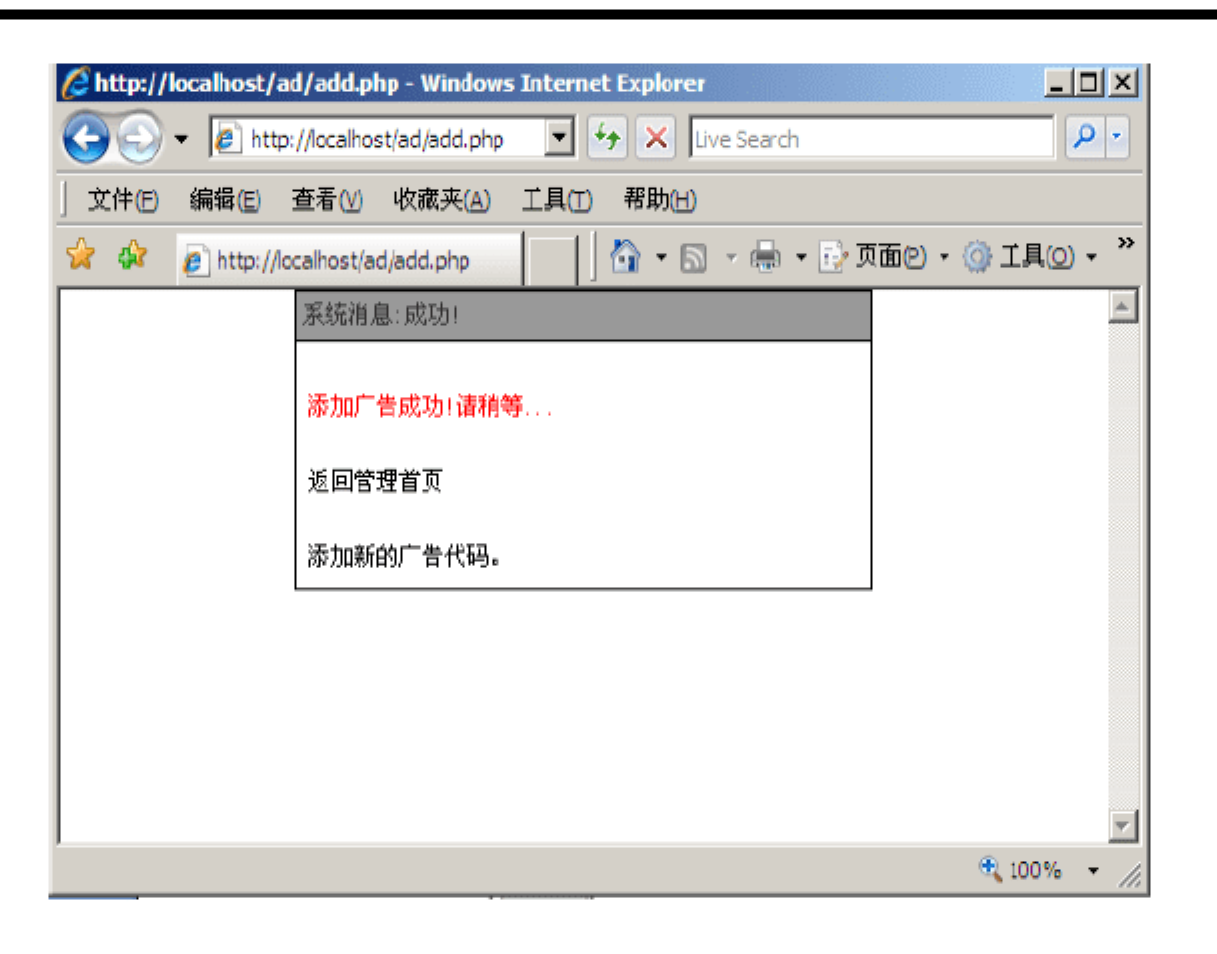


图 15.10　添加广告成功页面

15.5.2　广告生成与编辑操作

广告生成与编辑操作的原理都是对广告代码本身的一种修改（可以把广告代码生成看作是一次没有修改内容的编辑操作），编辑后按照指定规则生成 .js 后缀的广告文件。生成这个文件之后，用户就可以在自己的页面中嵌入调用代码，实现广告发布及内容管理。

1．广告生成

下面请看广告生成是如何实现的。具体的代码如下：

```
include_once(dirname(__FILE__)."/inc/include.smarty.php");
                                                    //加载 Smarty 配置文件
include(dirname(__FILE__)."/inc/include.function.php");   //加载函数库文件
require(dirname(__FILE__)."/config/config.php");          //加载容配置文件
$id=$_REQUEST[id];
$nx=$_REQUEST[nx];
$notice=$_REQUEST[notice];
$data=$_REQUEST[data];
$gge=$_REQUEST[gge];
$alljs=$_REQUEST[alljs];
/*批量生成 JS*/
if($_REQUEST[action]=="makejs"){                          //获得动作类型
echo "<xmp>";
$msg="";
$echoid="";
$tmp=explode("|www_allen|",file_get_contents($dbtable));  //截取字符串
$len=count($tmp);
//取得列数
/*所有 JS */
 if($alljs){
    for($i=0;$i<$len;$i++){
    $tmp[$i] = trim($tmp[$i]);
    if($tmp[$i] == "") continue;
    $info=explode("|allen|",$tmp[$i]);                    //切割字符串数组
    $echoid.="$addata/$info[0].js <br>";
    if($info[1]=="纯 HTML"){
    $data=allen_thu($info[2]);                  //执行正则表达式的搜索和替换
```

```
    allen file($data,"$addata/$info[0].js");
    }else{
    allen file($info[2],"$addata/$info[0].js");
    }
 }
 }else{
    for($i=0;$i<$len;$i++){
    $tmp[$i] = trim($tmp[$i]);
    if($tmp[$i] == "") continue;
    $info=explode("|allen|",$tmp[$i]);                    //切割字符串数组
    /*JS 的 ID*/
    $tcount=count($ REQUEST['idjs']);                     //计算 idjs 条数
    for($j=0;$j<=$tcount;$j++){
        $idjs=$ REQUEST['idjs'][$j];
        echo "$idjs\n";
            if($info[0]==$idjs){
                $echoid.="$addata/$info[0].js <br>";
                    if($info[1]=="纯 HTML"){
                        $data=allen thu($info[2]);
                        allen_file($data,"$addata/$info[0].js");
                        }else{
                        allen_file($info[2],"$addata/$info[0].js");
                        }
            }
        }
    }
}
isok("生成以下 ID 的 JS 成功：<p>$echoid");
}
```

【代码解读】

程序首先判断操作的类型 action=“makejs”代表生成对应的 JS 广告文件，还有一个作用就是，通过操作的类型来实现程序的流程控制，区分生成和修改的业务代码。判断操作代码如下：

```
if($_REQUEST[action]=="makejs"){
    //生成 JS 文件代码
}
```

生成的文件的代码如下：

```
allen_file($data,"$addata/$info[0].js");
```

上述代码中，allen_file()函数是在公共函数库中创建的，用来生成指定文件的函数。在广告列表页面选择要生成的广告条项目，单击“生成选定 Js”按钮完成生成任务，效果如图 15.11 所示。

通过之前的程序执行流程了解到，广告代码会生成对应的一个对应.js 后缀的文件，保存到根目录中的 js 文件夹中。打开这个文件夹，预览效果如图 15.12 所示。

2．广告编辑

广告编辑的作用是在广告内容调整时，可以通过该程序做出相应的修改。该程序的实现原理与广告生成部分类似，广告编辑的程序执行流程是通过调用判断操作类型，处理从表单中获得的修改信息，调用公共函数库中的 allen_file()函数完成广告内容的修改操作的。

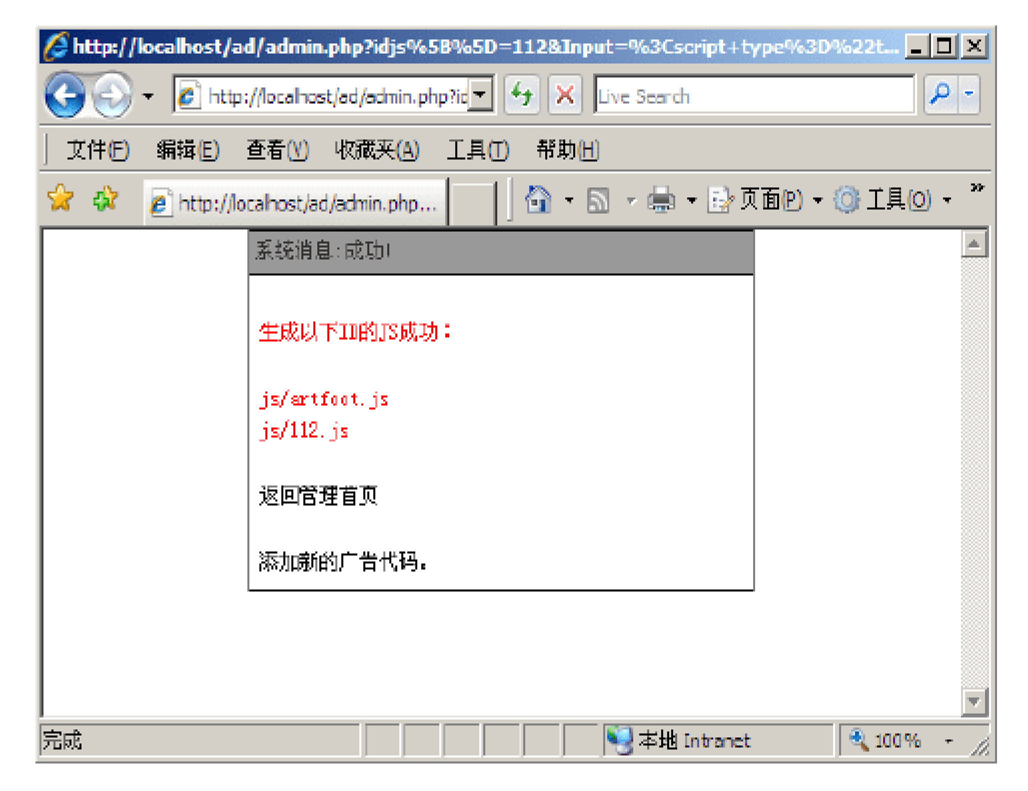


图 15.11　生成广告代码

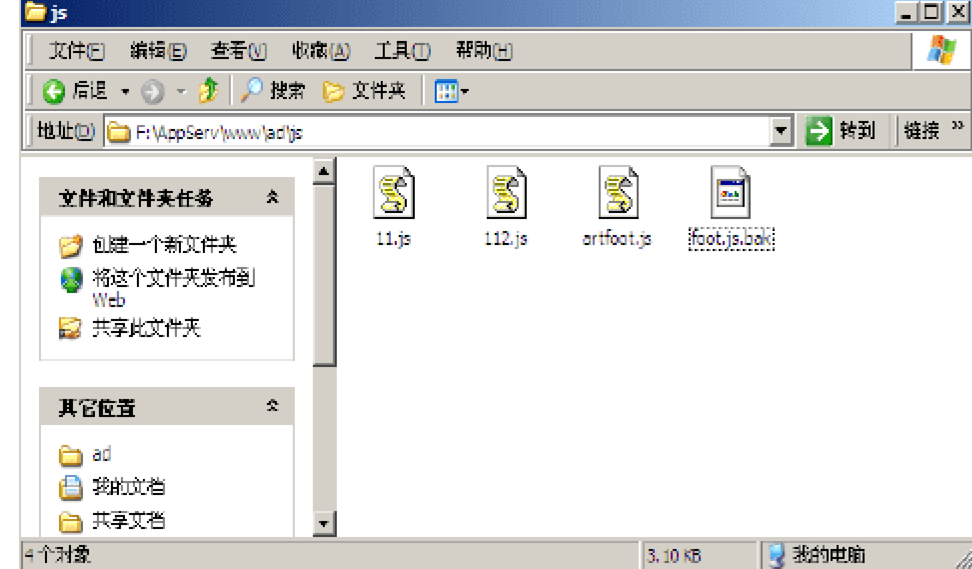


图 15.12　生成广告 JS 文件

广告编辑的实现代码如下：

```
if($_POST["action"]!=""){                    //动作流程控制
//die("$_POST[action] ggg");
$msg="";
$tmp=explode("|www_allen|",file_get_contents($dbtable));
$len=count($tmp);                            //取得列数
for($i=0;$i<$len;$i++){
$tmp[$i] = trim($tmp[$i]);
if($tmp[$i] == "") continue;
    $info=explode("|allen|",$tmp[$i]);
if($info[0]==$_POST[id]){                    //获得广告 id
                                             //处理生成内容
    $msgk=array($id,$nx,$data,$notice,$gge);
    $msgk2 = implode("|allen|",$msgk);
    $msg.=$msgk2."|www_allen|";
    continue;
}
$rows=$tmp[$i];
$msg.=$rows."|www_allen|";
}
//echo "\--$msg--";
$fp=fopen("$dbtable","w"); //打开指定文件“w+”（打开文件方式为可读写）
@fwrite($fp,$msg); //echo "成功!!<A href=?".$_SERVER['QUERY_STRING']."">跳
转</A>";
fclose($fp);
//判断类型
if($nx=="纯 HTML"){
    $data=allen_thu($data);
    allen_file($data,"$addata/$id.js");
    }else{
    allen_file($data,"$addata/$id.js");
    }
echo "<meta HTTP-EQUIV=\"REFRESH\" CONTENT='5;URL=admin.php?action=edit&
id=$_POST[id]'>
<center><br><br>修改该广告成功!<br>
<a href='admin.php?action=edit&id=$_POST[id]'>继续修改该 JS 文件</a> 
 
<a href='./'>返回 JS 管理首页 </a>";
exit;
```

```
//isok('修改广告成功!');
}
//操作类型为“编辑”时的后续操作
if($_GET[action]=="edit"){
$msgd=explode("|www_allen|",file_get_contents($dbtable));
$rows=count($msgd);
for ($i=0;$i<=$rows;$i++){
$tmp2=explode("|allen|",$msgd[$i]);
$msgn[$i]=$tmp2[0];
//echo $msgn[$i];
if(eregi("^$id$",$msgn[$i])){
$chj_nr=$msgd[$i];
}
}
if(!$chj_nr){error("好像没有该广告吧?");}
$tmp=explode("|allen|",$chj_nr);
}
//广告数组标签
$tmp[2]=str_replace("&","&amp;",StripSlashes("$tmp[2]"));
$smarty->assign("tmp", $tmp);
//加载管理模板
$smarty->display("./admin.html");
```

【代码解读】

上面代码的流程与之前介绍过的广告生成程序类似，这里需要注意的是，eregi()函数的功能是不区分大小写的正则表达式匹配。

根据广告编辑程序生成 Smarty 模板标签，来替换编辑页面的相应内容。替换后的核心的 HTML 页面代码如下：

```
<div align="left">
<table width="700" border="1" cellpadding="3" cellspacing="0"
    bgcolor="#CCCCCC">
    <tr>
        <td width="16%">广告 ID：<font color="#FF0000">*</font><br>
        <br>
        </td>
        <td width="84%"><input name="id" type="TEXT" class="input"
            id="id" value="{|$tmp[0]|}" size="25" maxlength="30" readonly>
        广告 ID 不能修改! </td>
    </tr>
    <tr>
        <td height="14">广告类型<font color="#FF0000">*</font> 
         </td>
        <td height="14"><select name="nx" id="nx">
            <option value="脚本式" {|if $tmp[1]=="脚本式"|}selected{|/if|}>
            脚本式</option>
            <option value="纯 HTML" {|if $tmp[1]=="纯 HTML"|}selected
            {|/if|}>纯 HTML</option>
        </select></td>
    </tr>
    <tr>
        <td>规格(宽*高)</td>
        <td><input name="gge" type="TEXT" class="input"
            value="{|$tmp[4]|}" size="20" maxlength="30"></td>
    </tr>
    <tr>
        <td>广告内容<font color="#FF0000">*</font></td>
```

```
        <td><textarea name="data" cols="80" rows="20" wrap="virtual"
            class="input" id="data" tabindex="2">{|$tmp[2]|}
</textarea> <br>
        提示：使用纯 HTML 代码时，因服务器配置不同，有些服务器不能自动将半角的“'”替
        换成“\'”，如果代码中含有单引号'，请先将代码中的 '替换成“\'”就可以了。</td>
    </tr>
    <tr>
        <td height="86" valign="top">
        <p>简单注释：</p>
        </td>
        <td valign="middle"><textarea name="notice" cols="80" rows="4"
            wrap="virtual" class="input" id="notice" tabindex="2">
            {|$tmp[3]|}</textarea>
        </td>
    </tr>
    <tr align="center">
        <td colspan="2"><input type="submit" name="action" value="修改"
            class="botton"> <input type="reset" value="重填" class="botton"
            name="reset"> <input type="button" value="返回上一页"
            onClick="javascript:history.back(-1);" class='botton'></td>
    </tr>
    <tr align="center">
        <td colspan="2">调用代码： <input name="Input2"
            onfocus="this.select()"
            value='&lt;script type=\"text/javascript\" src=\&quot;/js/
            {|$tmp[0]|}.js\&quot;&gt;&lt;/script&gt;'
            size="20" /></td>
    </tr>
</table>
</div>
```

在广告列表页面单击“管理”链接，或者在浏览器中输入如下形式链接，可以看到预览效果如图 15.13 所示。对广告内容做出修改后单击“修改”按钮，可以看到提示信息如图 15.14 所示。

```
http://localhost/ad/admin.php?action=edit&id=11
```

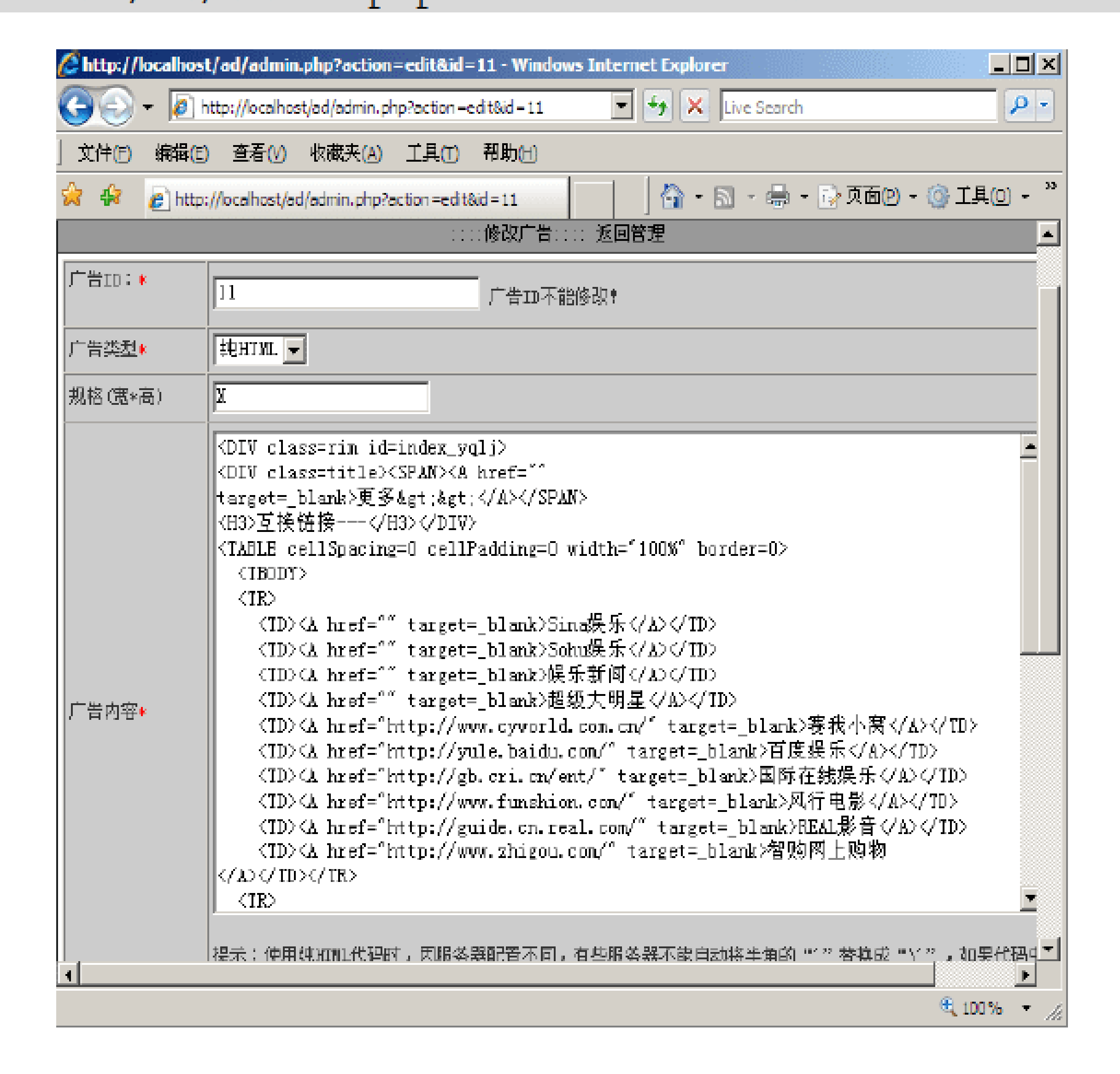


图 15.13　广告修改预览

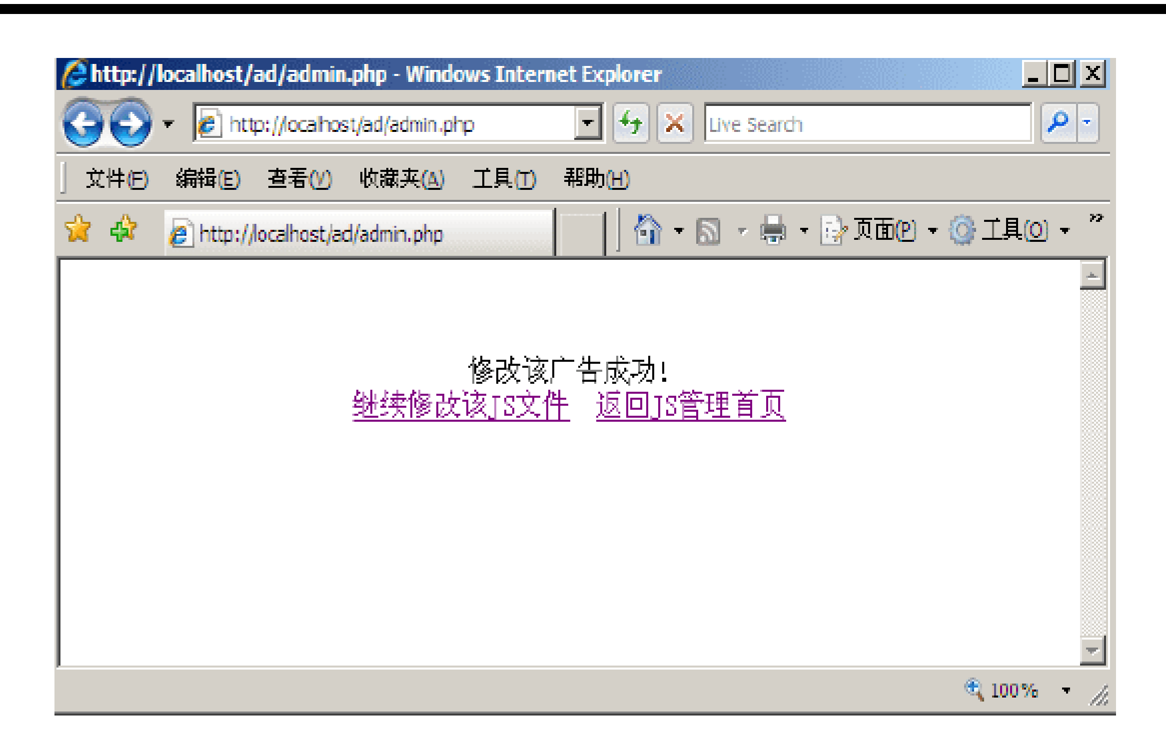


图 15.14　修改成功提示页面

15.5.3　广告删除操作

广告删除操作的作用是清除过期和错误的广告，不仅是在列表中清除而且要删除已经生成的广告文件。广告删除程序代码如下：

```
<?php
/*********************************************
'文件名：del.php
'主要功能：删除程序
'说明：判断 action 的动作(del.php?action=del&id=foot)
'*********************************************/
include once(dirname(  FILE  )."/inc/include.smarty.php");//加载 Smarty 配置文件
include(dirname(  FILE  ). "/inc/include.function.php");   //加载函数库文件
require(dirname(__FILE__)."/config/config.php");           //加载容配置文件
//判断本次的操作类型避免误操作
$id=$ REQUEST[id];                                          //获得广告 id
$nx=$_REQUEST[nx];
$notice=$ REQUEST[notice];                                  //获得提示信息
$data=$_REQUEST[data];                                      //获得内容数组
$gge=$ REQUEST[gge];
$alljs=$_REQUEST[alljs];
if($ POST[action]=="删除"){
$msg="";
$tmp=explode("|www allen|",file get contents($dbtable));
$len=count($tmp)-1;
for($i=0;$i<$len;$i++){
    $info=explode("|allen|",$tmp[$i]);
    if($info[0]==$ POST[id]){
        continue;
    }
    $tmp[$i]="$tmp[$i]|www allen|";
    $msg.=$tmp[$i];}
    $fp=fopen("$dbtable","w"); //打开指定文件“w+”（打开文件方式为可读写）
    @fputs($fp,$msg);
    unlink($addata."/".$id.".js");
    //echo $addata."/".$id.".js";
    isok('删除广告成功!');
    fclose($fp);                //关闭句柄链接
}
```

```
//生成该调广告的 id 标签
$smarty->assign("id", $id);
$smarty->display("./del.html");
?>
```

【代码解读】

删除操作分为两个步骤实现。首先是在广告存储文件中删除，使用重新写入的办法将原有的规则清除，代码如下：

```
$fp=fopen("$dbtable","w");//打开指定文件“w+”（打开文件方式为可读写）
@fputs($fp,$msg);
```

在存储文件中清除了之后还需要把已经生成的文件删除掉，这里使用 PHP 内置的 unlink()函数实现，它的功能是删除指定路径下的文件，完整的步骤代码如下：

```
$fp=fopen("$dbtable","w");//打开指定文件“w+”（打开文件方式为可读写）
@fputs($fp,$msg);
unlink($addata."/".$id.".js");
//echo $addata."/".$id.".js";
isok('删除广告成功!');
fclose($fp);
```

下面看删除程序的 HTML 页面代码。其中只给出了表单的部分，完整的代码请读者参考 tpl 文件夹下的 del.html 文件内容。代码如下：

```
<form action="del.php?action=del" method="post">
<table cellspacing=1 cellpadding=3 width=50% bgcolor="#000000"
    class="fonts" align="center" name="passwd">
    <tr bgcolor="#99CC66">
        <td height="11" align="center" bgcolor="#999999" class="mycss">
        <div align="center" class="style1"><> 删除广告</div>
        </td>
    </tr>
    <tr bgcolor="#FFFFFF">
        <td align="center" class="mycss">
        <div align="center">
        <p><br>
        本操作并不能删除包含该代码的网页上的代码，建议不要删除。</p>
        <p>您确定删除吗??</p>
        </div>
        <div align="center"><br>
        <input type="hidden" name=id value="{|$id|}"> <input
            type="submit" name="action" value="删除" class="botton"> <input
            type="button" value="不删除，返回上一页"
            onClick="javascript:history.back(-1);" class='botton'> <br>
        </div>
        </td>
        </tr>
    </table>
    </form>
```

执行删除程序效果如图 15.15 所示。单击“删除”按钮后，执行广告被彻底删除，预览效果如图 15.16 所示。

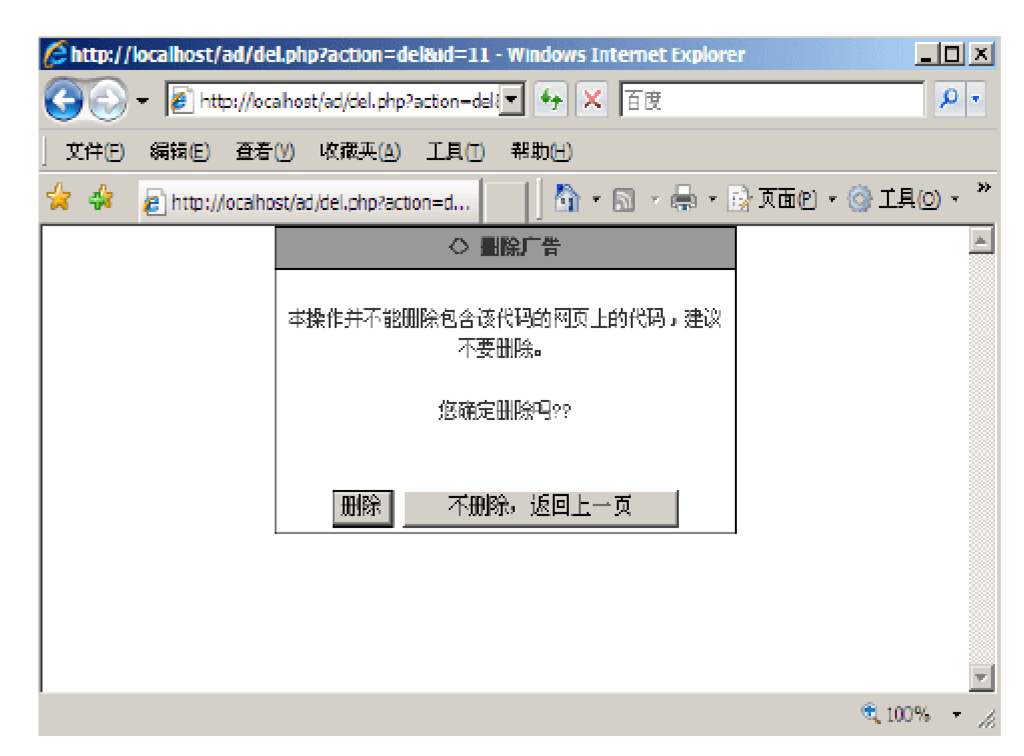


图 15.15　删除确认页面

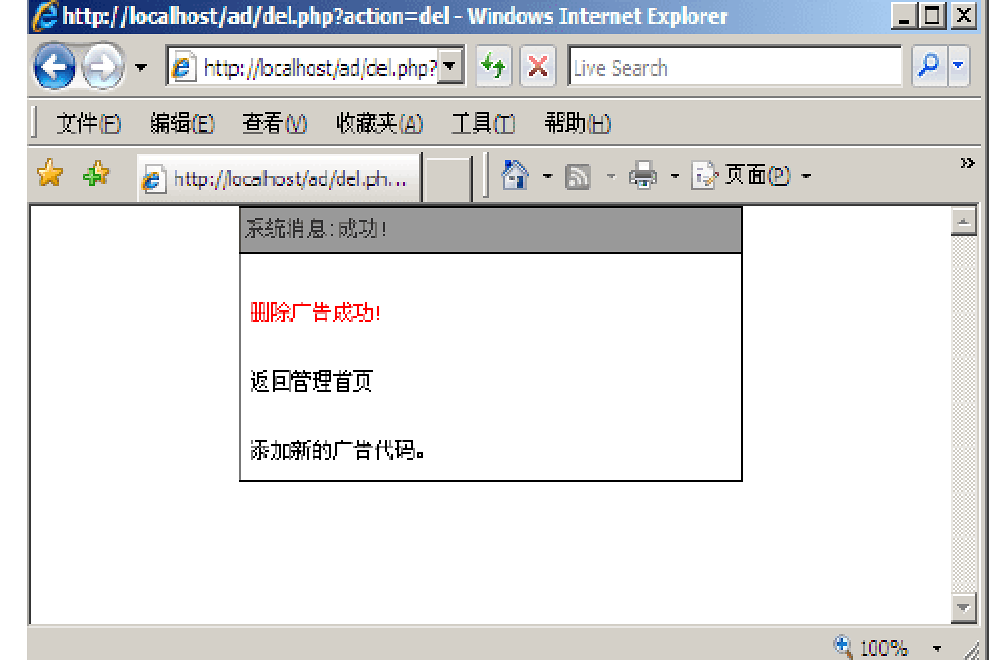


图 15.16　删除成功页面

15.6　小　　结

广告管理模块是一个典型的 PHP 管理 JavaScript 文件的应用。因为广告本身有其特殊性，所以在管理时采用文档和标签的存储模式，可以最大程度地提供灵活性，便于添加、修改。

本章从第 15.1 节广告管理概述讲起，首先介绍了广告管理基本特征和广告系统的工作流程。第 15.2 节介绍了广告系统的文件目录结构设计及说明，详细介绍了各个文件和文件夹的作用及需要完成的功能。根据功能需求在第 15.3 节介绍了核心的功能函数。第 15.4 节进入代码实现部分，介绍了显示广告列表程序和内容搜索程序。第 15.5 节介绍了广告管理的操作程序，包括新增广告操作、广告生成与编辑操作、广告删除操作。通过本章的学习，读者可以对 PHP 管理 JavaScript 文件这类的应用有整体和直观的认识。

第 16 章　文件管理器（文件处理+Smarty）

文件管理器模块是介绍 PHP 对文件和目录操作管理的程序模块，PHP 中内置了丰富的操作文档和目录的函数，它们的结合实现了文件的管理功能。读者学习本章，要注意总结程序功能框架（如获取文件内容、编辑指定文件等）。

本章主要涉及的知识点如下。

- 文件管理的常用函数：PHP 内置了支持文件管理的函数，按照操作对象可以分为目录操作类、文件操作类和字符串处理类。
- 文件句柄：句柄是一类特殊的智能指针。
- section 循环：是 Smarty 模板引擎中定义的循环函数标签。
- getcwd()函数：PHP 内置的获得文件当前路径的函数。

16.1　文件管理器概述

文件管理器是对文档进行可视化管理的工具程序，它可以执行一些常见的文档操作，便于多个用户的操作和文档管理。

16.1.1　文件管理器的基本特征

文件管理器提供了一个访问文件和应用程序的集中访问点，一个完整的文件管理器需要完成下列的功能：

- 创建文件夹和文档；
- 显示文件和文件夹；
- 打开指定路径的文件夹；
- 编辑文件内容并保存；
- 文件重命名；
- 删除指定文件和文件夹。

具备了上述的文件管理功能的程序就可以被称作文件管理器，也是文件管理的核心功能和基本特征，其他的功能都是核心的辅助和扩展。

16.1.2　工作流程描述

文件管理器是基于 B/S 结构的 PHP 文档操作程序，是 PHP 对文件和文件夹操作管理

的典型应用。下面介绍文件管理的工作流程，程序的工作流程如图 16.1 所示。

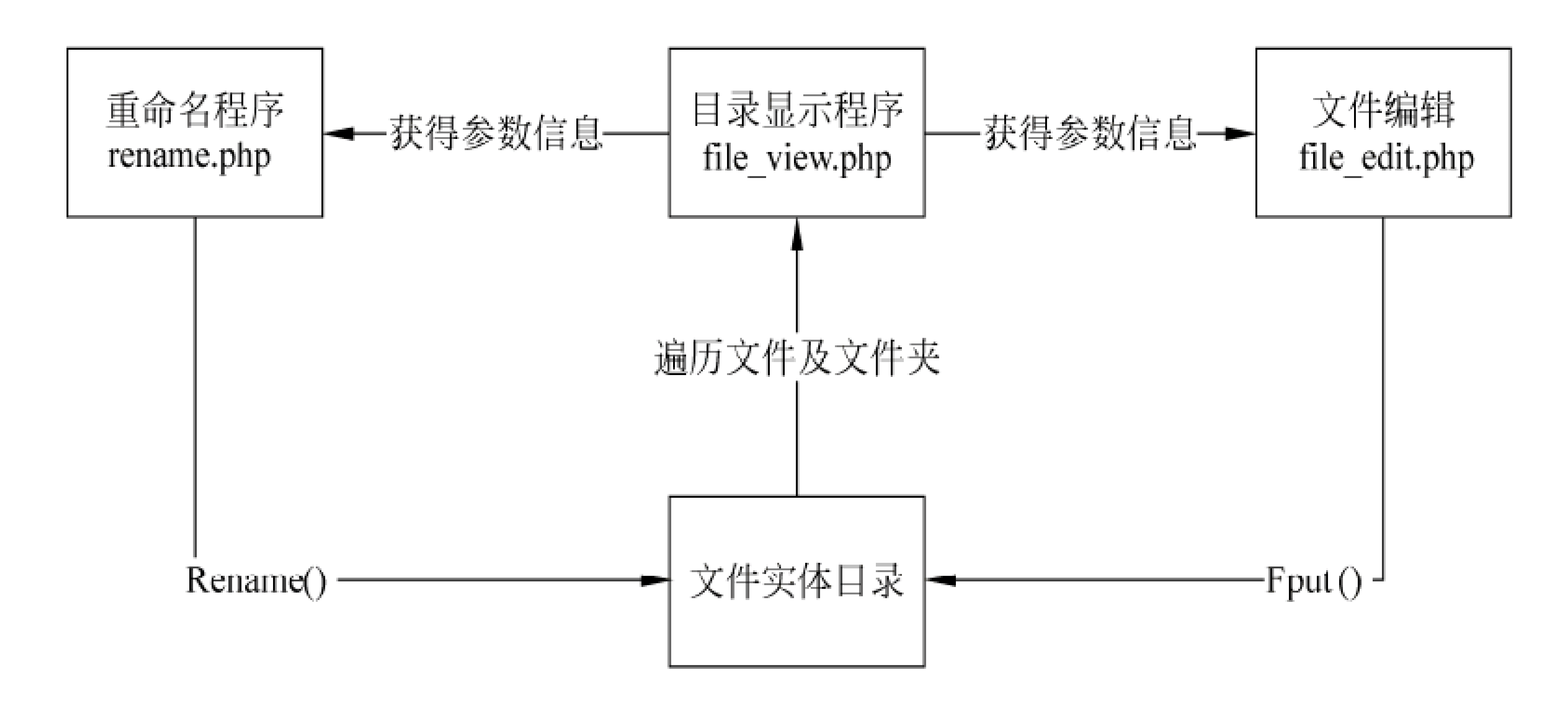


图 16.1　程序工作流程图

16.2　文件处理函数说明

PHP 中提供了丰富的操作文档和目录的函数，这些内置函数涵盖了绝大部分的文档类操作。在实际的应用中，这些函数通常组合使用完成一个功能，引入这些常用函数极大地提高了开发效率。本节将结合文件管理器模块，介绍开发中出现频率较高的文件处理函数。

16.2.1　目录类函数

对目录操作的函数在实际应用中使用频率非常高，常用来定义一些基础的变量信息，比如获得目录地址、创建句柄等。下面介绍几个常用的目录函数。

1．chdir()函数

【函数说明】

```
bool chdir ( string directory )
```

【作用】 改变目录，将当前目录改为 directory。
【参数】 directory（改变后的新目录）。
【返回值】 如果成功，返回 true；失败，则返回 false。
【例子】

```
<?php
//getcwd()说明
echo getcwd() . "\n";//输出当前目录
chdir('webroot');
echo getcwd() . "\n";//输出改变后的当前目录
?>
```

上述代码执行后结果如下：

```
/home/user
```

```
/home/user/webroot
```

2．opendir()函数

【函数说明】

```
int opendir(string path);
```

【作用】 opendir()函数用来打开目录资料流，返回的整数是可供其他目录函数操作的句柄（handle）。

【参数】 path（要打开的目录路径）。

【返回值】 如果成功，返回目录句柄的资源；失败，则返回 false。

【例子】

```
<?php
    //opendir()函数说明
    $dir = opendir("images");  //打开 images 目录
    //列出 images 目录中的文件
    while (($file = readdir($dir)) !== false)
    {
        echo "filename: " . $file . "<br />";
    }
    closedir($dir);              //关闭目标句柄
?>
```

上述代码执行后结果如下：

```
filename: .
filename: ..
filename: cat.gif
filename: dog.gif
filename: food
filename: horse.gif
```

3．readdir()函数

【函数说明】

```
string readdir ( resource dir_handle )
```

【作用】 返回目录中下一个文件的文件名。

【参数】 dir_handle（目录句柄的句柄资源，由 opendir()函数创建）。

【返回值】 如果成功，返回文件名；失败，则返回 false。

【例子】

```
<?php
//readdir 函数说明
if ($handle = opendir ( '.' )) {                //判断是否为根目录
    while ( false !== ($file = readdir ( $handle )) ) {
                                                //返回目录中下一个文件的文件名
        if ($file != "." && $file != "..") {
            echo "$file\n";
        }
    }
```

```
    closedir ( $handle );                    //关闭文件句柄
}
?>
```

上述代码的作用是，列出当前目录的所有文件并去掉“.”和“..”。

4．scandir()函数

【函数说明】

```
scandir(directory,sort,context)
```

【作用】 返回目录中下一个文件的文件名。

【参数】

- directory（必选）：规定要扫描的目录。
- sort（可选）：规定排列顺序，默认是 0（升序），如果是 1 则为降序。
- context（可选）：规定目录句柄的环境。

【返回值】 成功，返回包含有文件名的 array；如果失败，则返回 false。如果 directory 不是个目录，则返回布尔值 false，并生成一条 E_WARNING 级的错误。

【例子】

```
<?php
//scandir 函数说明
$dir = '/tmp';                 //设置目录名称
$files1 = scandir($dir);     //列出指定路径中的文件和目录
$files2 = scandir($dir, 1);
//输出测试
print_r($files1);
print_r($files2);
?>
```

上述代码执行后结果如下：

```
Array
(
    [0] => .
    [1] => ..
    [2] => bar.php
    [3] => foo.txt
    [4] => somedir
)
Array
(
    [0] => somedir
    [1] => foo.txt
    [2] => bar.php
    [3] => ..
    [4] => .
)
```

16.2.2　文件操作函数

获得某个文件的基本信息或者对该内容的操作，在 PHP 中只能依靠内置的函数实现。

因此，了解和熟悉这些功能函数非常重要。下面介绍对文件本身操作的相关函数，读者在实际的使用中应该注意函数的变量形式和返回值。

1. 文件内容操作函数

在文件操作函数中有几个函数，通常会结合使用完成打开文件、获取内容、修改文件内容的操作。下面通过两个例子了解文件内容获取、文件内容修改是怎么实现的。

【例子 1】 文件内容获取：

```
<?php
$handle = @fopen("/tmp/readme.txt", "r");  //创建文件句柄
if ($handle) {
    while (!feof($handle)) {                //测试文件指针是否到了文件结束的位置
        $buffer = fgets($fd, 4096);         //从文件指针中读取一行
        echo $buffer;
    }
    fclose($handle);                        //关闭已打开的文件指针
}
?>
```

【代码解读】

上面代码的作用是逐行读取文件中的内容，再将内容串赋值或者直接输出，就完成了文件内容的获取。

【例子 2】 文件内容修改：

```
<?php
$filename = 'readme.txt';   //设置文件名称
$somecontent = "添加这些文字到文件\n";
//判断文件存在并且可写
if (is writable ( $filename )) {
    /*添加模式打开$filename，文件指针将会在文件的开头，使用 fwrite()的时候，
      $somecontent 将要写入的地方*/
    if (! $handle = fopen ( $filename, 'a' )) {
        echo "不能打开文件 $filename";
        exit ();
    }
    // 将$somecontent 写入到已打开的文件中
    if (fwrite ( $handle, $somecontent ) === FALSE) {
        echo "不能写入到文件 $filename";
        exit ();
    }
    echo "成功地将 $somecontent 写入到文件$filename";
    fclose ( $handle );
} else {
    echo "文件 $filename 不可写";
}
?>
```

常见的文件内容操作函数见表 16.1。

2. filesize()函数

【函数说明】

```
int filesize ( string filename )
```

表 16.1　常见的文件内容操作函数含义表

函数名	参　　数	返　回　值	描　　述
fopen()	filename、mode	true/false	打开文件或者 URL
fgets()	length	true/false	从文件指针中读取一行
fwrite()	handle、length	成功返回写入的字符数，出现错误时则返回 false	写入文件（可安全用于二进制文件）
fclose()	handle	成功则返回true，失败则返回 false	关闭一个已打开的文件指针

【作用】 取得文件大小。

【参数】 filename（目标文件名）。

【返回值】 返回文件大小的字节数，如果出错，返回 false 并生成一条 E_WARNING 级的错误。

【例子】

```
<?php
//指定目标文件名
$filename = 'readme.txt';
echo $filename . ': ' . filesize($filename) . ' bytes';
?>
```

上述代码执行后，输出结果类似：

```
somefile.txt: 1024 bytes
```

3．filemtime()函数

【函数说明】

```
int filemtime ( string filename )
```

【参数】 filename（目标文件名）。

【返回值】 返回文件上次被修改的时间，出错时返回 false。时间以 Unix 时间戳的方式返回。

【例子】

```
<?php
    //指定目标文件名
    $filename = 'readme.txt';
    if (file_exists($filename)) {       //检查文件是否存在
        echo "$filename was last modified: " . date ("F d Y H:i:s.",
        filemtime($filename));
    }
?>
```

上述代码执行后，输出结果类似：

```
readme.txt was last May: December 29 2010 22:16:23.
```

4．readfile()函数

【函数说明】

```
int readfile ( string filename [, bool use_include_path [, resource
context]] )
```

【作用】 读入一个文件并写入到输出缓冲。

【参数】 filename（目标文件名）。

【返回值】返回从文件中读入的字节数，如果出错，返回 false。但是可以通过@readfile()形式调用来抑制报错，否则会显示错误信息。

【例子】

```
<?php
$filename = 'readme.txt';
    echo readfile($filename);//输出目标文件
?>
```

上述代码执行后，输出结果类似：

```
There are two lines in this file.
This is the last line.
```

16.2.3　字符串处理函数

字符串的处理在任何一种编程语言中都是不可或缺的。字符串处理的特点是操作繁复，很多过程都是重复出现，幸好 PHP 内置了字符串处理函数，可以满足大多数情况的需求，下面来介绍几个出现频率较高的函数。

1. strrpos()函数

【函数说明】

```
strrpos(string,find,start)
```

【作用】 函数查找字符串在另一个字符串中最后一次出现的位置。

【参数】

- string（必需）：规定被搜索的字符串。
- find（必需）：规定要查找的字符。
- start（可选）：规定开始搜索的位置。

【返回值】 如果成功，返回位置，否则返回 false。

【例子】

```
<?php
//定义一个字符串变量
$str="Hello WORLD!";
echo strtoupper($str);
?>
```

上述代码执行后输出结果为：

```
HELLO WORLD!
```

2. strtoupper()函数

【函数说明】

```
string strtoupper ( string string )
```

【作用】 字符串全转为大写。

【参数】 string（待转换的字符串）。

【返回值】 该函数将字符串 str 全部变大写字符串。

注意：该函数对大小写敏感。

【例子】

```
<?php
    echo strrpos("Hello world!","wo");//输出大写字符
?>
```

上述代码执行后输出结果为：

```
6
```

3．substr()函数

【函数说明】

```
string substr(string string, int start, int [length]);
```

【作用】 本函数将字符串 string 的第 start 位起的字符串取出 length 个字符。若 start 为负数，则从字符串尾端算起。若可省略的参数 length 存在，但为负数，则表示取到倒数第 length 个字符。

【参数】

- string（必需）：规定要返回其中一部分的字符串。
- start（必需）：规定在字符串的何处开始。
- length（可选）：规定要返回的字符串长度。默认是直到字符串的结尾。

【返回值】 返回截取后的字符串。

注意：如果 start 是负数并且 length 小于等于 start，则 length 为 0。

【例子】

```
<?
echo substr("abcdef", 1, 3);
echo substr("abcdef", -2);
echo substr("abcdef", -3, 1);
echo substr("abcdef", 1, -1);
?>
```

上述代码执行后输出结果为：

```
bcd
ef
d
bcde
```

4．htmlentities()函数

【函数说明】

```
htmlentities (string,quotestyle,character-set);
```

【作用】 将所有的字符都转成 HTML 字符串。

【参数】

- string（必需）：规定要转换的字符串。
- quotestyle（可选）：规定如何编码单引号和双引号。
- character-set（可选）：字符串值，规定要使用的字符集。

【返回值】 将所有 string 的字符都转成 HTML 的特殊字集字符串返回。

注意：无法被识别的字符集将被忽略，并由 ISO-8859-1 代替。

【例子】

```
<html>
<body>
<?php
    $str = "John & 'Adams'";                       //为字符变量赋值
    echo htmlentities($str, ENT_COMPAT);    //将所有的字符都转成 HTML 字符串
    echo "<br />";
    echo htmlentities($str, ENT_QUOTES);
    echo "<br />";
    echo htmlentities($str, ENT_NOQUOTES);//将所有的字符都转成 HTML 字符串
?>
</body>
</html>
```

在浏览器中输出上面的代码：

```
John & 'Adams'
John & 'Adams'
John & 'Adams'
```

如果在浏览器中查看源代码，会看到如下 HTML 标签：

```
<html>
<body>
John &amp; 'Adams'<br />
John &amp; &#039;Adams&#039;<br />
John &amp; 'Adams'
</body>
</html>
```

5. eregi_replace()函数

【函数说明】

```
string eregi_replace(string pattern, string replacement, string string);
```

【作用】 字符串比对解析并取代，与大小写无关。

【返回值】 匹配字符串。

【例子】

```
<?php
$string = "This is a test";
echo str_replace(" is", " was", $string);        //字符串比对解析并取代
echo "<br>";
echo ereg_replace("( )is", "\\1was", $string); //字符串比对解析并取代
```

```
echo "<br>";
echo ereg_replace("(( )is)", "\\2was", $string);//字符串比对解析并取代
?>
```

上述代码执行后输出结果为：

```
This was a test
This was a test
This was a test
```

6．str_replace()函数

【函数说明】

```
str_replace(find,replace,string,count);
```

【作用】 str_replace()函数使用一个字符串替换字符串中的另一些字符。

【参数】

- find（必需）：规定要查找的值。
- replace（必需）：规定替换 find 中的值的值。
- string（必需）：规定被搜索的字符串。
- count（可选）：一个变量，对替换数进行计数。

【返回值】 返回替换后的字符串。

注意：该函数是二进制传输。

【例子】

```
<?php
echo str_replace("world","John","Hello world!");//目标字符替换
?>
```

上述代码执行后输出结果为：

```
Hello John!
```

16.3　文件管理器

文件管理器的核心功能是显示和管理文件和文件夹。本节将介绍显示文件和文件夹的程序实现和查看，显示文件和文件夹包括文档模式和图片浏览模式。在实际应用中，需要根据不同的需求选择合适的显示模式，下面具体介绍这两种模式。

16.3.1　文件和目录浏览

文件和目录的浏览程序的设计思路是使用目录类函数和文档函数获取文件和目录的基本信息，再根据这些信息将目录中的内容配置后遍历出来，同时配合路径的操作就可以完整地实现文件和目录浏览了。

创建文件名为 file_view.php 的 PHP 脚本文件，该脚本文件的功能是生成目录的链接、

循环读取目录中的目录及文件，并生成对应的模板标签输出到页面模板中。该文件的具体代码如下：

```
<?php
/*******************************************
'文件名：file view.php
'主要功能：显示文件目录程序
'说明：
 *******************************************/
//加载 Smarty 配置文件
include once (dirname (   FILE   ) . "/inc/include.smarty.php");
//获取用户单击页面上的目录链接生成的新的目录信息
if (! isset ( $ GET [currentdir] ) || empty ( $ GET [currentdir] )){
    $dir = getcwd ();
}else{
    $dir = $ GET [currentdir];
}
chdir ( $dir );              //改变目录
$current = "当前目录:" . getcwd () . "<br>";
$current dir = getcwd ();
$dh = opendir ( $dir );      //打开目录
//循环读取目录中的目录及文件
$i = 0;
while ( $item = readdir ( $dh ) ) {
                             //如果是目录
    if (is_dir ( $item )) {
                             //对当前目录
        if ($item == ".") {
            $currentdir = getcwd ();
            $info [$i] ['name'] = "<a href=$PHP SELF?currentdir=" .
            $currentdir . "><IMG height=16 src=\"./image/dir2.gif\"
            width=16 align=absMiddle border=0>根目录</A>";
            $info [$i] ['item'] = $item;
        }                    //对上一级目录
        else if ($item == "..") {
            $currentdir = getcwd () . "\\..";
            $info [$i] ['name'] = "<a href=$PHP SELF?currentdir=" .
            $currentdir . "><IMG height=16 src=\"./image/dir2.gif\"
            width=16 align=absMiddle border=0>上级目录</A>";
            $info [$is] ['item'] = $item;
        } else               //对子目录
{
            $currentdir = getcwd () . "\\$item";
            $info [$i] ['name'] = "<a href=$PHP SELF?currentdir=" .
            $currentdir . "><IMG height=16 src=\"./image/dir.gif\" width=16
            align=absMiddle border=0>" . $item . "</A>";
            $info [$i] ['item'] = $item;
        }
    } //如果是文件
    else {
        //截取文件后缀
        $extname = substr ( $item, strrpos ( $item, "." ) );
        //显示 txt 和 php 文件的链接信息，通过链接可以打开文件
        if (strtoupper ( $extname ) == ".PHP" || strtoupper ( $extname ) ==
        ".TXT") {
            switch (strtoupper ( $extname )) {
```

```
                case ".css" :
                    $gif = "css.gif";
                    break;
                case ".PHP" :
                    $gif = "php.gif";
                    break;
                default :
                    $gif = "txt.gif";
            }
            $currentdir = getcwd ();
            $info [$i] ['name'] = "<a href=./show file.php?currentdir=" .
            $currentdir . "&filename=" . $item . "&" . $type . "=" .
            $extname .">
     <IMG height=16 src=\"./image/" . $gif . "\" width=16 align=absMiddle
     border=0>" . $item . "</A>";
            $info [$i] ['item'] = $item;
        } //对于其他类型的文件，只给出文件名
        else {
            //echo $item;
            $info [$i] ['name'] = "<a href=''>
     <IMG height=16 src=\"./image/txt.gif\" width=16 align=absMiddle
     border=0>" . $item . "</A>";
            $info [$i] ['item'] = $item;
        }
    }
    //显示文件或目录的其他信息
    //文件类型和大小
    if (is dir ( $item ))
        $file_size = "目录";
    else
        $file size = round ( filesize ( $item ) ) . "KB";
    $info [$i] ['size'] = $file size;
    //echo "<td>$file_size</td>";
    //文件创建时间
    $create_date = date ( "y-m-d h:i:sA", filectime ( $item ) );
                                    //filectime 取得文件创建的时间
    echo "<td>$create_date";
    //文件最后修改时间
    $update_date = date ( "y-m-d h:i:sA", filemtime ( $item ) );
                                    //filemtime 取得文件最后修改的时间
    //echo "<td>$update_date</td>";
    $info [$i] ['update date'] = $update date;
    $i ++;
}
closedir ( $dh ); //关闭目录\
//生成信息标签
$smarty->assign ( "info", $info );
//当前文件路径
$smarty->assign ( "current dir", $current dir );
$smarty->assign ( "current", $current );
$smarty->display ( "./file view.html" );
?>
```

【代码解读】

梳理 file_view.php 程序的流程，可以了解到程序的执行步骤如下：

（1）获得当前目录信息，通过 opendir()函数创建文件操作句柄，代码如下：

```
//获取用户单击页面上的目录链接生成的新的目录信息
if(!isset($_GET[currentdir])||empty($_GET[currentdir]))$dir=getcwd();
```

```
   else
     $dir=$ GET[currentdir];
chdir($dir);            //改变目录
$current="当前目录:".getcwd()."<br>";
$current dir=getcwd();
$dh=opendir($dir);      //创建目录操作句柄
```

（2）获得基本目录信息后，使用 while()函数循环遍历数组，然后通过 readdir()函数，循环读取目录中的目录名称及文件名称，基本的功能模型如下：

```
while($item = readdir($dh)){
    //对获得的数组处理操作…
}
```

细心的读者可能注意到了，获得的数组有些是文件，有些是文件夹。获得它们信息的操作是完全不同的，所以需要在上面的基本功能模型中对获得文件的类型进行判断，并根据不同的类型生成不同的数组，请看判断部分的代码：

```
//如果是目录
if(is dir($item))
{
    //对当前目录
    if($item==".")
    {
     $currentdir=getcwd();
     $info[$i]['name']="<a href=$PHP SELF?currentdir=".$currentdir.">
     <IMG height=16 src=\"./image/dir2.gif\" width=16 align=absMiddle
     border=0>根目录</A>";
    $info[$i]['item']=$item;
    }
  //对上一级目录
    else
    if($item=="..")
    {
     $currentdir=getcwd()."\\..";
     $info[$i]['name']="<a href=$PHP SELF?currentdir=".$currentdir.">
     <IMG height=16 src=\"./image/dir2.gif\" width=16 align=absMiddle
     border=0>上级目录</A>";
    $info[$is]['item']=$item;
    }
    else    //对子目录
    {
     $currentdir=getcwd()."\\$item";
     $info[$i]['name']="<a href=$PHP_SELF?currentdir=".$currentdir. ">
     <IMG height=16 src=\"./image/dir.gif\" width=16 align=absMiddle
     border=0>".$item."</A>";
     $info[$i]['item']=$item;
    }
}
//如果是文件
else
{
    //截取文件后缀
    $extname=substr($item,strrpos($item,"."));
    //显示 txt 和 php 文件的链接信息，通过链接可以打开文件
    if(strtoupper($extname)==".PHP"||strtoupper($extname)==".TXT")
    {
     //配置文件显示图片
```

```
    switch (strtoupper($extname)) {
        case ".css":
        $gif="css.gif";
        break;
        case ".PHP":
        $gif="php.gif";
        break;
        default:
        $gif="txt.gif";
    }
      $currentdir=getcwd();
     $info[$i]['name']="<a href=./show_file.php?currentdir=".
     $currentdir."&filename=".$item."&".$type."=".$extname.">
     <IMG height=16 src=\"./image/".$gif."\" width=16 align=absMiddle
     border=0>".$item."</A>";
     $info[$i]['item']=$item;
    }
    //对于其他类型的文件，只给出文件名
    else
    {
     //echo $item;
     $info[$i]['name']="<a href=''>
     <IMG height=16 src=\"./image/txt.gif\" width=16 align=absMiddle
     border=0>".$item."</A>";
     $info[$i]['item']=$item;
    }
```

【代码解读】

这里需要注意的是，通过截取字符串的方式获得文件的后缀名的代码如下：

```
$extname=substr($item,strrpos($item,"."));
```

然后根据不同的文件后缀名配置不同的显示图片，这里使用 switch()函数也是为了方便新增显示图片的类型。这里请读者仔细体会 switch()函数作为流程控制和条件匹配的不同特点。

（3）获得该目录下文件的辅助信息并组合上面循环遍历获得数组，生成模板标签，输出到模板页面。获得文件信息和生成模板标签的代码如下：

```
//显示文件或目录的其他信息
//文件类型和大小
if(is dir($item))$file size="目录";
else
   $file size=round(filesize($item))."KB";
   $info[$i]['size']=$file size;
   //echo "<td>$file size</td>";
//文件创建时间
$create date=date("y-m-d h:i:sA",filectime($item));
                                  //filectime ---   取得文件创建的时间
//echo "<td>$create date";
//文件最后修改时间
$update date=date("y-m-d h:i:sA",filemtime($item));
                                  //filemtime ---   取得文件最后修改的时间
//echo "<td>$update date</td>";
$info[$i]['update date']=$update date;
$i++;
}
closedir($dh);   //关闭目录
//var_dump ($info);
```

```
//生成信息标签
$smarty->assign("info", $info);
//当前文件路径
$smarty->assign("current dir", $current dir);
$smarty->assign("current", $current);
$smarty->display("./file_view.html");
```

【代码解读】

这里需要注意的是，作为良好的编程习惯，在对文件操作结束后需要使用 close()函数关闭目录句柄。

创建文件名为 file_view.html 的 HTML 页面文件，该模板的作用是通过模板引擎驱动，将模板标签中的内容数组输出到模板中，实现文件管理器的文件和目录浏览，显示部分的代码如下：

```
<TABLE class=tbtitle style="BACKGROUND: #e2f5bc" cellSpacing=1
    cellPadding=1 width="96%" align=center border=0>
    <TBODY>
        <TR height=24>
            <TD align=middle width="28%" bgColor=#edf9d5><STRONG>文件名
            </STRONG></TD>
            <TD align=middle width="16%" bgColor=#edf9d5><STRONG>文件大小
            </STRONG></TD>
            <TD align=middle width="22%" bgColor=#edf9d5><STRONG>最后修改
            时间</STRONG></TD>
            <TD align=middle width="34%" bgColor=#edf9d5><STRONG>操作
            </STRONG></TD>
        </TR>
        <TR bgColor=#ffffff height=22>
            <TD>{|$current|}</TD>
            <TD colSpan=3><A style="COLOR: red" href="">[图片浏览器]</A>
            </TD>
        </TR>
        {|section name=infoname loop=$info|}
        <TR onMouseMove="javascript:this.bgColor='#F9FBF0';"
            onmouseout="javascript:this.bgColor='#FFFFFF';"
bgColor=#ffffff
            height=22>
            <TD>{|$info[infoname].name|}</TD>
            <TD>{|$info[infoname].size|}</TD>
            <TD align=middle>{|$info[infoname].update_date|}</TD>
            <TD><a
                href="./file edit.php?currentdir={|$current dir|}
                &filename={|$info[infoname].item|}">[编辑]</A>
              <A href="./rename.php?rep=&fic={|$info[infoname].
            item|}">[改名]</A>
              <A href="">[删除]</A>   <A href="">[移动]</A></TD>
        </TR>
        {|/section|}
        <TR>
            <TD align=left bgColor=#bee8fc colSpan=4 height=24><A href=
            "">[根目录]</A>
              <A href="">[新建文件]</A>   <A href="">[新建目录]</A>
              <A
                href="">[文件上传]</A>   <A href="">[空间检查]</A>
                  </TD>
        </TR>
    </TBODY>
```

```
</TABLE>
```

【代码解读】

内容信息被输出到模板，使用到了 Smarty 模板引擎中使用的遍历输出函数，循环体样式如下：

```
{|section name=infoname loop=$info|}
<TR onMouseMove="javascript:this.bgColor='#F9FBF0';"
    onmouseout="javascript:this.bgColor='#FFFFFF';" bgColor=#ffffff
    height=22>
    <TD>{|$info[infoname].name|}</TD>
    <TD>{|$info[infoname].size|}</TD>
    <TD align=middle>{|$info[infoname].update_date|}</TD>
    <TD><a
        href="./file edit.php?currentdir={|$current dir|} &filename=
        {|$info[infoname].item|}">[编辑]</A>
      <A href="./rename.php?rep=&fic={|$info[infoname].item|}">[改
    名]</A>
      <A href="">[删除]</A>   <A href="">[移动]</A></TD>
</TR>
{|/section|}
```

【代码解读】

通过使用 section 循环标签，在循环体内的所有内容全部被循环输出，当然也包括 HMTL 代码。因此，重复的 HTML 在模板中只被使用了一次，这就是引入模板可以减少页面代码量的直接原因。但是，在减少代码量的同时也增加了程序处理部分的工作量，因为需要照顾模板标签的输出规则。

在浏览器中执行下面的链接：

```
http://localhost/fmanager/file_view.php
```

文件管理器的文件和目录的浏览效果，如图 16.2 所示。



图 16.2　文件和目录的浏览

16.3.2　文件查看程序

文件查看程序是文件管理的最基本的功能，它的设计要点是针对获取到的文件格式特点做相应的 HTML 标签替换，这里用到了 readfile()函数和 htmlentities()函数。读者通过这个文件查看程序，可以体会 readfile()函数和 htmlentities()函数的区别和不同的适用领域。

创建文件名为 show_file.php 的 PHP 脚本文件，该文件的代码如下：

```
<?php
//获取文件信息
$currentdir = $ GET [currentdir];
$filename = $ GET [filename];
$type = $ GET [type];
//显示 PHP 文件内容
if (strtoupper ( $type ) == ".PHP") {
    readfile ( $currentdir . "\\" . $filename );
} else {
    //显示 txt 文件内容
    $fp = fopen ( $currentdir . "\\" . $filename, "r" );
    while ( $line = fgets ( $fp ) ) {
        $line = htmlentities ( $line, ENT COMPAT, "utf-8" );
        echo $line;
    }
    fclose ( $fp );
}
?>
```

在目录浏览页面点击一个 PHP 文件或者 txt 文件图标的链接，也可以在浏览器中执行下面的链接：

```
http://localhost/fmanager/show file.php?currentdir=F:\AppServ\www\fmana-
er&filename=file_view.php&=.php
```

文件管理器的文件和目录的浏览效果，如图 16.3 所示。

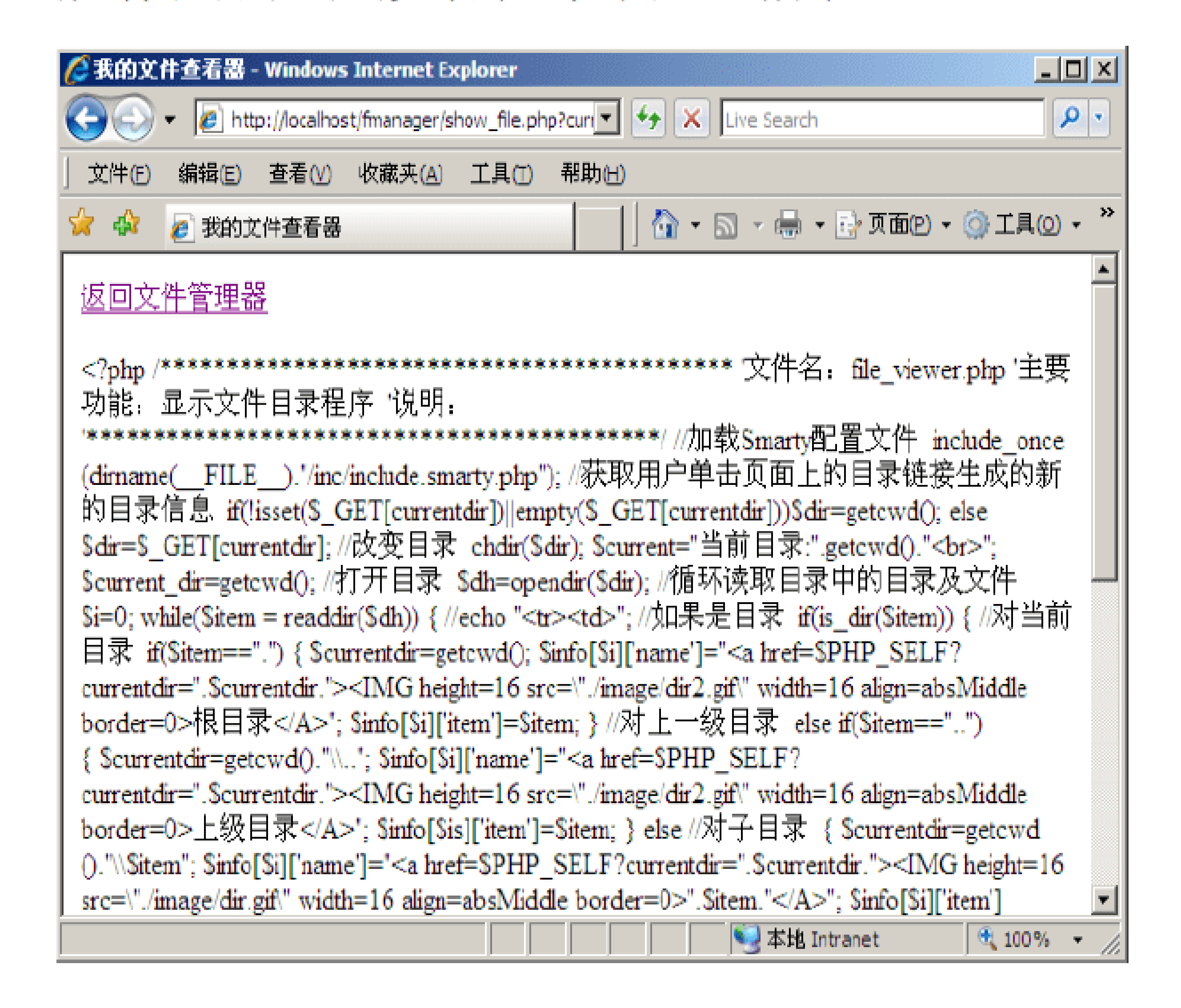


图 16.3　文件查看

16.4　文件管理功能

前一节介绍了如何通过浏览器浏览当前的文件和目录。当然，一个完善的文件管理程序需要有对文件和目录的管理功能。本章将介绍常见的几个文件管理功能，这些功能与上文介绍过的浏览、查看功能共同构成了一个完整的文件管理器。

16.4.1　文件编辑

文件编辑功能在实际中的使用频率很高。通常每种文件格式都有对应的编辑软件，这类软件功能非常强大，但在实际应用中绝大多数的需求只需改变文件中的某一段内容。因此使用文件管理器中的文件编辑功能可以很方便地实现查看、编辑、保存等最常见的文档操作。为了使读者更清晰地了解文档编辑的流程，将这部分拆分成几个独立的程序分别完成。

1. 编辑内容

编辑的先决条件是要获取到文件的内容，并合理地组织这些元素，直观地展示给使用者。创建文件名为 file_edit.php 的 PHP 脚本文件，该文件将获取文件中的内容，处理之后生成模板标签输出到模板页面中。具体的代码如下：

```
<?php
/*******************************************
'文件名：file_edit.php
'主要功能：文件编辑程序
'说明：
 *******************************************/
//加载 Smarty 配置文件
include_once (dirname ( __FILE__ ) . "/inc/include.smarty.php");
//获取文件信息
$activepath = $_GET [currentdir];                          //获得当前目录
$filename = $_GET [filename];                              //获得文件名称
$type = $_GET [type];                                      //获得操作类型
if (! isset ( $backurl ))
    $backurl = "";
    $activepath = str_replace ( "..", "", $activepath );   //字符串替换
    $filename = str_replace ( "..", "", $filename );
    $file = "$activepath/$filename";
    $content = "";
    $fp = fopen ( $file, "r" );
    while ( $line = fgets ( $fp ) ) {
        $content .= htmlentities ( $line, ENT_COMPAT, "utf-8" );
        //规则过滤
        $content = eregi_replace ( "<textarea", "< textarea", $content );
        $content = eregi_replace ( "</textarea", "< /textarea", $content );
        $content = eregi_replace ( "<form", "< form", $content );
        $content = eregi_replace ( "</form", "< /form", $content );
    }
fclose ( $fp );
```

```
$contentView = "<textarea name='str' style='width:100%;height:400'>" .
$content . "</textarea>\r\n";
$GLOBALS ['filename'] = $filename;
//文件名标签
$smarty->assign ( "filename", $filename );
$smarty->assign ( "activepath", getcwd () );
$smarty->assign ( "content", $content );
$smarty->display ( "./file_edit.html" );
?>
```

【代码解读】

上面的代码使用了常用的文件内容获取模式 fopen()、while()和 fgets()结合获取文档内容的模式。使用代码表述如下：

```
$fp=fopen($file,"r");
while($line=fgets($fp)) {
    $content.=htmlentities($line,ENT_COMPAT,"utf-8");
                                          //将字符都转成 HTML 字符串
    //规则过滤
    //生成制定规则的数组
}
```

【代码解读】

读者可能注意到了，上面的代码中生成模板标签的部分替换了部分的 HTML 代码，这是为了避免文档内容中的代码与模板中的代码冲突，所以做了必要的替换。因此在页面模板的设计时也需要注意这个问题。创建文件名为 file_edit.html 的 HTML 模板页面，该文件的功能是文件编辑的模板页面。具体的代码如下：

```
<table width="98%" border=0 cellpadding=0 cellspacing=0
    bordercolor=#111111 style="border-collapse: collapse">
    <form method="POST" action="file_save.php" name=form1
        onSubmit="return Post()"><input type="hidden" name="action"
        value="save"> <input type="hidden" name="fic"
        value="{|$filename|}"> <input type="hidden" name="rep"
        value="{|$activepath|}">
    <tr>
        <td width=78% valign="top">
        <table border=0 cellpadding=0 cellspacing=0
            style="border-collapse: collapse" width=100%>
            <tr>
                <td width=16% height="25">工作目录：</td>
                <td width="84%">    <input name=activepath
                    size=40 value="{|$activepath|}"> （空白表示根目录 ，不允
                    许用 “..” 形式的路径）</td>
            </tr>
            <tr>
                <td width=16% height="25">文件名称：</td>
                <td>    <input name=filename size=40
                    value="{|$filename|}">（不允许用“..”形式的路径）<input
                    type="button" name="Submit" value="使用可视化模式"
                onClick="location='file_manage_view.php?fmdo=
                editview&activepath=/dedecms&filename=baidunews.xml';"
                    class='nbt'></td>
            </tr>
        </table>
        </td>
```

```
    </tr>
    <tr>
        <td width=78% height="31" valign="top"><textarea name='code'
            style='width: 100%; height: 400'> {|$content|} </textarea></td>
    </tr>
    <tr>
        <td width=78% valign="top">
        <p align=center><input type=submit value="  保 存  " name=B1
            class='nbt'>   <input type=reset value="取消修改" name=B2
            class='nbt'>   <input type=button value="不理返回" name=B4
            onClick="javascript:history.go(-1);" class='nbt'> <br>
        </td>
    </tr>
    </form>
</table>
```

在目录浏览页面单击一个 PHP 文件或者 txt 文件的“编辑”选项，也可以在浏览器中执行下面的链接：

```
http://localhost/fmanager/file_edit.php?currentdir=F:\AppServ\www\fmana-
ger%20&filename=baidu.xml
```

文件管理器的文件编辑页面效果，如图 16.4 所示。

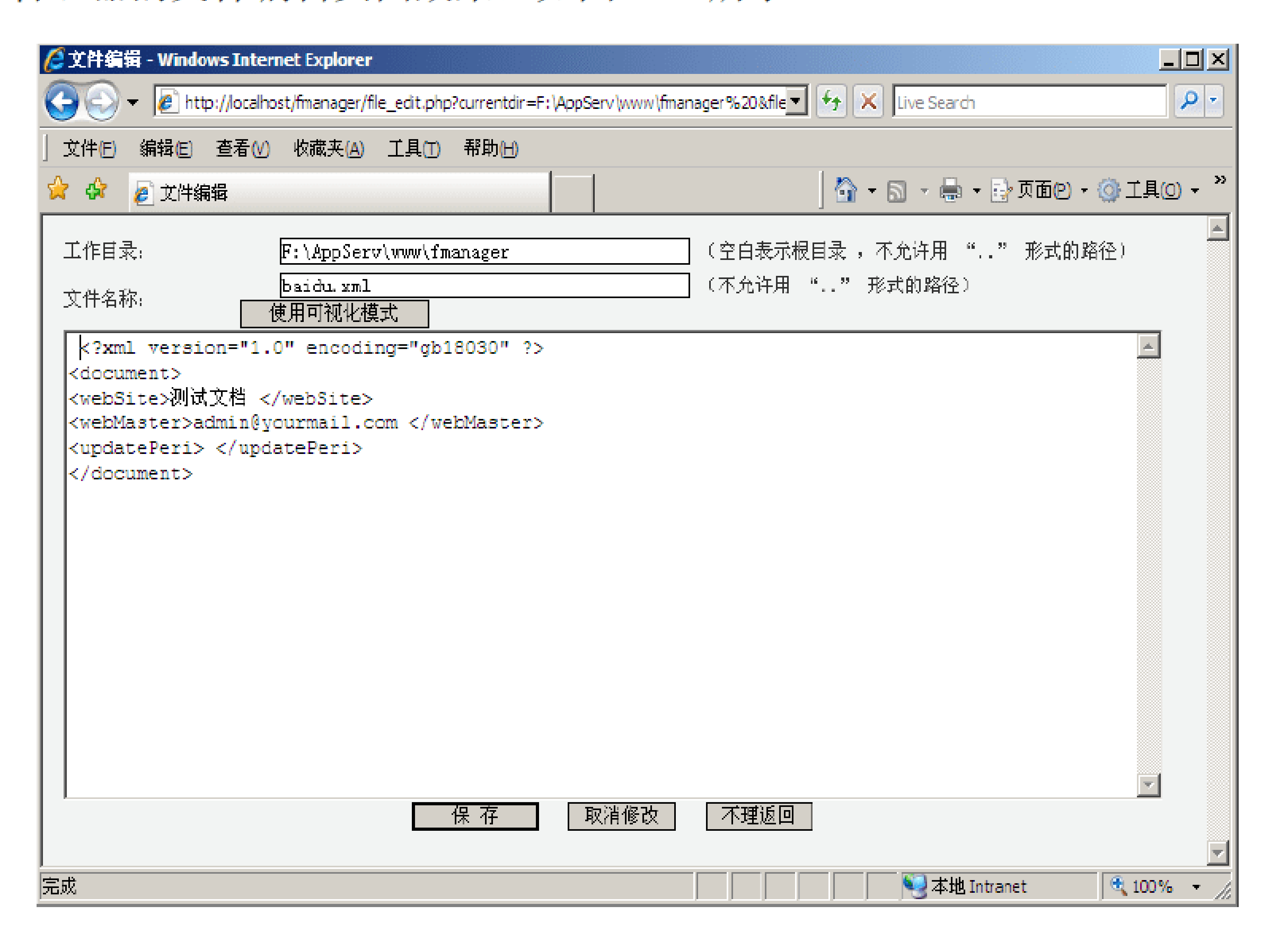


图 16.4　文件编辑

从图 16.4 中可以看到，文件编辑程序已经正确地读取了文件的内容和相关的信息，并通过编辑区域显示出来。

2. 文档保存

编辑后的文件需要再重新写入到源文件中，这就完成了文档编辑和保存。文档保存程序的设计思路如下。

（1）正确地接收表单传递过来的数据，打开 file_edit.html 文件看表单标签的名字，用到的标签如下：

```
<form method="POST" action="file save.php" name=form1
    onSubmit="return Post()"><input type="hidden" name="action"
    value="save"> <input type="hidden" name="fic"
    value="{|$filename|}"> <input type="hidden" name="rep"
    value="{|$activepath|}"> <input name=activepath size=40
    value="{|$activepath|}"> <input name=filename size=40
    value="{|$filename|}"> <textarea name='code'
    style='width: 100%; height: 300'> {|$content|} </textarea> <input
    type=submit value="  保 存  " name=B1 class='nbt'> <input
    type=reset value="取消修改" name=B2 class='nbt'> <input type=button
    value="不理返回" name=B4 onClick="javascript:history.go(-1);"
    class='nbt'>
```

（2）了解了对应表单传递的信息后，书写以下的代码段用来接收表单值并将内容写回文件中，代码如下：

```
$rep=getcwd();                  //获取当前目录
$fic=stripslashes($fic);        //去掉反斜线字符
$code=stripslashes($code);      //去掉反斜线字符
//switch 流程控制
switch($action) {
    case "save";
    $file = "$rep\$fic";
    $fp=fopen($fic,"w");
    fputs ($fp,$code);
    fclose($fp);
    //$output="提示信息";
    echo $output;
    break;
}
```

【代码解读】

程序判断表单的动作变量，当动作值为“save”时执行保存操作，这里依然是使用 PHP 内置的文件操作函数。可以归纳出一个将内容（字符串）“写入”指定文档的程序模型，模型的构成代码如下：

```
$file = "带路径的文件名";
$fp=fopen($file,"w");
fputs ($fp,写入内容);
fclose($fp);
```

（3）当保存完成之后，用户无法得知文档的状态，需要给用户一个提示信息，显示该次操作的信息，并返回先前的操作页面。这里使用$output 变量直接向浏览器中输出 HTML 页面代码，提示信息代码如下：

```
$output="
<html>
<head>
<title>系统提示</title>
<meta http-equiv=\ "Content-Type\" content=\ "text/html; charset=utf-8\ "
/>
<base target=' self' />
</head>
<body leftmargin='0' topmargin='0'>
<center><script>
      var pgo=0;
```

```
      function JumpUrl(){
        if(pgo==0){ location='file view.php'; pgo=1; }
      }
document.write(\"<br/><div style='width:400px;padding-top: 4px;height:
24;font-size:10pt;border-left:1px  solid  #b9df92;border-top:1px  solid
#b9df92;border-right:1px solid #b9df92;background-color:#def5c2;'>提示信
息：</div>\");
document.write(\"<div style='width:400px;height:100;font-size:10pt;
border:1px solid #b9df92;background-color:#f9fcf3'><br/><br/>\");
document.write(\"成功保存".$fic."文件！\");
document.write(\"<br/><br/><a href='file view.php'>如果你的浏览器没反应，请点
击这里...</a><br/><br/></div>\");
setTimeout('JumpUrl()',5000);</script></center>
</body>
</html>
";
echo $output;
break;
```

（4）在文档编辑页面修改文档内容后，单击“保存”按钮，出现提示信息如图 16.5 所示。

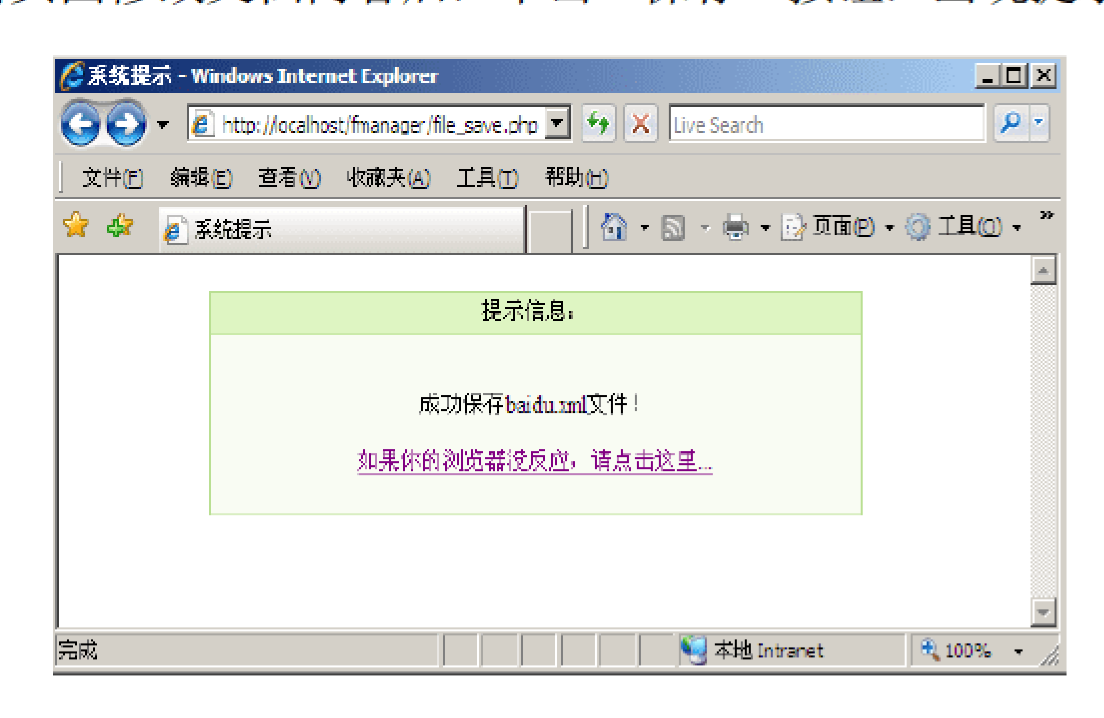


图 16.5　保存编辑文档

16.4.2　文件更名

文件管理中另一个常用的功能就是为文件重命名，使用的频率也非常的高。在 PHP 中，是使用 rename()函数来实现文档的重新命名的。创建文件名为 rename.php 的 PHP 脚本文件，程序会先获得路径信息处理后，再结合 rename.php 函数完成文档的重新命名。该文件的代码如下：

```
<?
/*******************************************
'文件名：rename.php
'主要功能：文件重命名
'说明：
'*******************************************/
//加载 Smarty 配置文件
include once(dirname(  FILE  )."/inc/include.smarty.php");
$racine=".";                      /*文件执行到的目录*/
$rep=getcwd();                    //获得当前目录
$fic=stripslashes($fic);          //非法字符过滤（反斜线）
$nom_fic=basename($fic);
```

```
$fic_new=$_POST['fic_new'];
function slash()
    {
    global $racine;
    if(ereg("\\\\",$racine)){$slash="\\";}
    else {$slash="/";}
    return $slash;
    }
                                                    //动作流程控制
switch($action) {
    case "rename";
    $old=$racine.slash().$fic;
    $new=dirname($old).slash().$fic_new;
    $fic_new=stripslashes($fic_new);        //非法字符过滤（反斜线）
    $fic_new=stripslashes($fic_new);
    $fic_new=str_replace("'","",$fic_new);//字符替换
    $fic_new=str_replace("\"","",$fic_new);
    $fic_new=str_replace("&","",$fic_new);
    $fic_new=str_replace(",","",$fic_new);
    $fic_new=str_replace(";","",$fic_new);
    $fic_new=str_replace("/","",$fic_new);
    $fic_new=str_replace("\\","",$fic_new);
    $fic_new=str_replace("`","",$fic_new);
if($fic_new=="")
{
     echo $message="异常";
}else if(file_exists($fic_new))
    {
    echo $message='文件名重复！';
    }
else
    {
    rename($old,$new);
    echo $message='成功更改文件名！';
    }
break;
default;
$message="异常错误";
break;
}
//模板变量设置
$smarty->assign("filename", $fic);
$smarty->assign("activepath", getcwd());
$smarty->display("./rename.html");
?>
```

【代码解读】

上面代码使用了 swith()函数做流程控制。具体地说，就是通过判断 action 的值来控制程序流程。通过 str_replace()函数过滤掉非法的命名方式，否则会导致程序无法正确打开。再通过 file_exits()函数判断文件名是否存在，避免被错误覆盖或者程序异常报错。最后使用 rename()函数完成重命名功能。

创建文件名为 rename.html 的 HTML 模板页面，用来显示重命名相关的信息，部分代码如下：

```
<form name='myform' method='POST' onSubmit='return CheckSubmit();'
    action='rename.php'><input type="hidden" name="action"
    value="rename"> <input type="hidden" name="fic"
```

```
    value="{|$filename|}"> <input type="hidden" name="rep"
    value="{|$activepath|}">
<table width='100%' border='0' cellpadding='1' cellspacing='1'
    align='center' class='tbtitle' style='background: #E2F5BC;'>
    <tr>
        <td colspan='2'><font color='#666600'><b>更改文件名，当前路径:
        {|$activepath|}</b></font></td>
    </tr>
    <tr>
        <td width='25%'>旧名称: </td>
        <td width='75%'><input name='oldfilename' type='input'
            id='oldfilename' size='40' value='{|$filename|}'></td>
    </tr>
    <tr>
        <td width='25%'>新名称: </td>
        <td width='75%'><input name='fic new' type='input' size='40'
            id='newfilename'></td>
    </tr>
    <tr>
        <td colspan='2'>
        <table width='270' border='0' cellpadding='0' cellspacing='0'>
            <tr align='center'>
                <td width='90'><input name='imageField1' type='image'
                    class='np' src='image/button ok.gif' width='60'
                    height='22'border='0' style='border: 0px'></td>
                <td width='90'><a href='#'><img class='np'
                    src='image/button reset.gif' width='60' height='22'
                    border='0'onClick='this.form.reset();return false;
                    ' style='border: 0px'></a></td>
                <td><a href='#'> <img src='image/button back.gif'
                    width='60' height='22' border='0' onClick='history.
                    go(-1);'style='border: 0px'></a></td>
            </tr>
        </table>
        </td>
    </tr>
</table>
</form>
```

在文件和目录浏览页面选择文件名为“baidu.xml”，点击“改名”链接，效果如图 16.6 所示。



图 16.6　更改文件名

在更改文件名页面，输入新文件名“baidunew.xml”，单击“确定”按钮，成功后程序会返回目录浏览页面，如图 16.7 所示。从图中可以看到，文件名已经变成了“baidunew.xml”，重命名成功了。

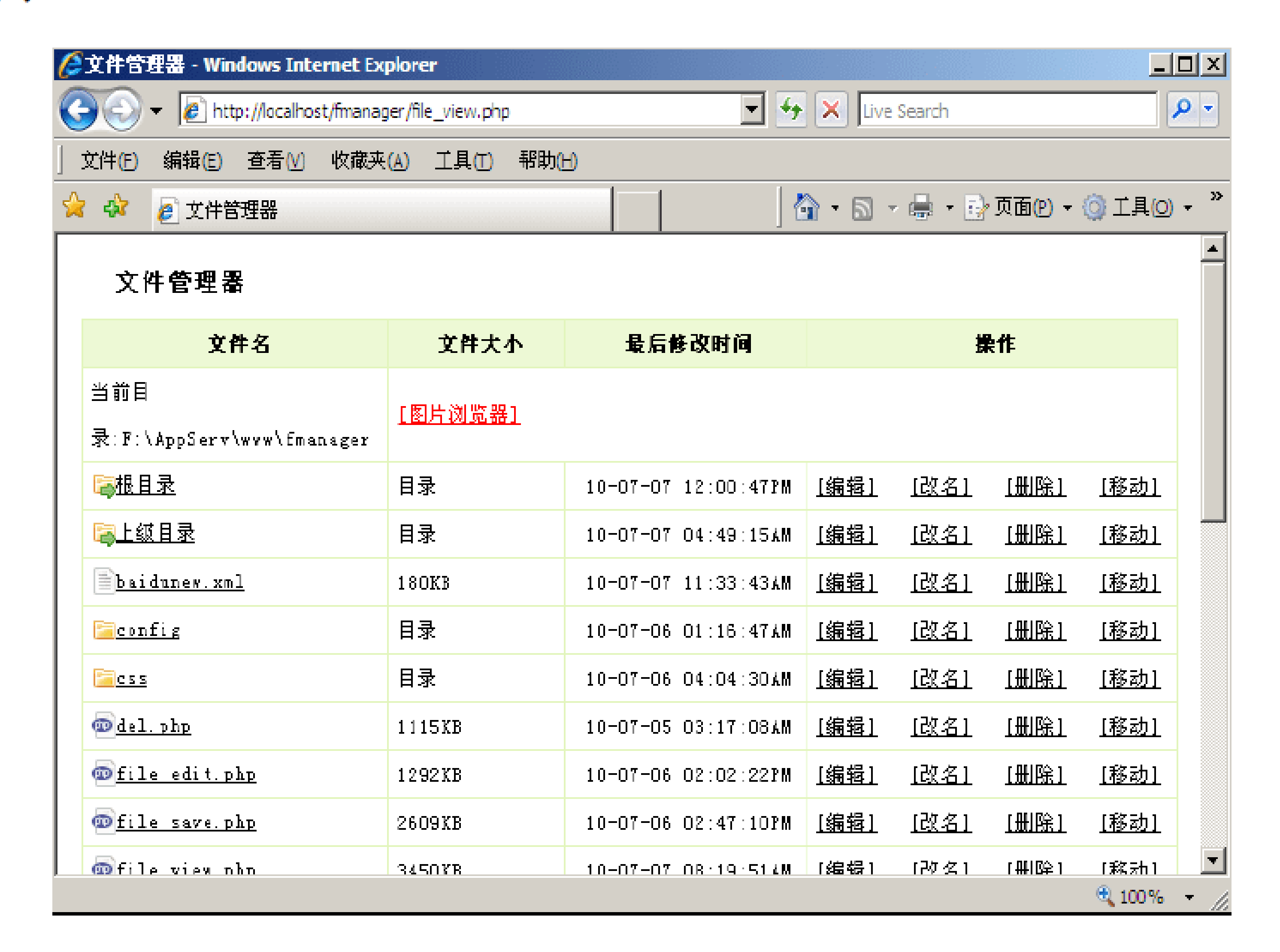


图 16.7　文件目录浏览页

16.5　小　　结

文件管理器模块的核心是理解 PHP 在文档和目录方面的处理逻辑。本章第 16.1 节介绍了文件管理器的基本概念，随后简要地介绍了文件管理器的基本特征和工作流程描述。第 16.2 节系统地介绍了 PHP 处理文件的函数，一共分成 3 类，分别是目录类函数、文件操作函数和字符串处理函数，这 3 类搭配使用完成对文件的管理功能。第 16.3 节是使用这些函数的具体的实践，完成文件和目录浏览功能和文件查看功能。第 16.4 节介绍了两个常用的文件管理功能，即文件编辑功能和文件更名功能。

第 3 篇　PHP 项目开发实战

第 17 章　内容管理系统（OOP+URL Rewrite+HTML 编辑器）

内容管理系统（CMS）是使用 PHP 搭建完整管理系统的典型应用，本章首先介绍 CMS 的设计基础，然后逐一介绍搭建内容管理系统所使用的基础类库，再逐一添加管理功能。本章采用面向对象（OOP）编程方式，读者在学习的过程中注意体会，如何在 PHP 中使用面向对象设计（Object Oriented Programming，OOP）的开发方式编程。

本章主要涉及的知识点如下。

- CMS：是 Content Management System 的缩写，意思是“内容管理系统”。
- 页面“伪静态”化：页面“伪静态”化是通过 URL 重写来实现页面键接的伪静态化（URL Rewrite 技术）。
- PHP 中的类：类是变量与作用于这些变量的函数的集合，被广泛地应用在面向对象编程设计（OOP）的开发模式中。
- PHP 中的 copy 函数：PHP 中的 copy()函数通常被用在附件的上传功能。

17.1　CMS 设计基础

CMS（Content Management System，内容管理系统），从命名中就可以了解到 CMS 的核心功能就是对网站的内容进行管理（编辑和发布），一个完善的 CMS 可以使网站建设避免从零开始设置数据库结构和后台程序的开发，保证网站的建设周期，同时大大地节约了各种成本。

17.1.1　CMS 基本概念

在 CMS 的设计中广泛地使用了模板的概念，在设计和使用模板前必须要理解下面几个概念。

（1）板块（封面）模板：指网站主页或比较重要的栏目封面频道使用的模板，本模块采用“index_识别 ID.php”命名。例如，网站首页的模板为 index.php，这里并没有采用模板引擎的方式，而是使用 PHP 脚本将内容输出到模板。

（2）列表模板：指网站某个栏目的所有文章列表的模板。本模块采用“list_c_栏目 ID.html”命名。

（3）档案模板：表示文档查看页的模板，如文章模板，本模块采用“show_c_栏目 ID_文章 ID.html”命名。

（4）其他模板：一般系统常规包含的模板有登录模板、用户注册模板、搜索条模板、栏目导航模板、网站底栏模板等。此外，需要新增功能也可以自定义一个模板，创建为任意文件。

17.1.2　CMS 主要实现功能

CMS 系统是以管理网站内容为设计目的，按照功能的重要程度可以分为核心功能和辅助功能。核心功能完成了 CMS 的基础管理功能，辅助功能是根据网站不同的业务需要提供的辅助功能，通常会以插件的方式提供给用户选择安装。

1．核心功能

- 文章编辑：Web 编辑器是内容管理的最重要的工具，编辑器提供给编辑人员方便地发布和修改网站内容，常见的编辑器（如 FCKeditor）提供了丰富和强大的编辑功能，可以完全满足日常的发布和维护需要。本章将使用一个轻量级的内容管理器，逻辑更为清晰，便于读者理解内容编辑器的实现原理。
- 静态 URL：友好的、静态的 URL 不仅使得搜索引擎更容易抓取网站内容，而且也方便了用户访问。静态化按照实现方式可以分为两种，第一种是通过引擎的方式直接将内容生成真正的 HTML 页面；第二种是所谓的“伪静态”，现在通过 URL Rewrite 技术已经可以很容易实现这个功能，对所有 CMS 来说，这是必需的一个功能。这里需要注意的一点是，并非只有 HTML 后缀才是唯一的静态 URL，任何不含查询字符串的 URL，不管是什么后缀扩展名，都是静态 URL。
- 全站搜索：CMS 系统一般使用后台数据库存储网站内容，这让全站搜索变得易于实现。让访问者快速找到想要的内容是 CMS 的首要任务，这是一个 CMS 系统必要的功能。
- 栏目导航菜单：网站栏的目的随着业务调整会有相应的调整，这涉及栏目对应内容的组织，并生成导航菜单。这些自动生成的导航菜单提供给访问者更有效地找到网站的内容。这个功能对所有 CMS 系统都是非常必要的。

2．辅助功能

- 模板管理：模板是 CMS 将内容和页面展示分离的机制，模板的引入是为了实现模板风格的调整（改版）不影响内容的展示。完善的模板系统可以通过可视化参数调整模板的风格样式，并且可以方便的在不同的风格模板之间切换，不影响前台的内容展示。
- 站点地图：站点地图将全站内容的静态链接，按照逻辑关系生成在一个页面中，这主要是为了方便搜索引擎抓取网站的内容。生成网站地图（Website Map）功能通常是通过插件程序的方式安装在 CMS 系统中的。
- 版本控制：熟悉程序开发的流程的读者应该都知道版本控制的重要性，版本控制是一个完善的 CMS 系统必须拥有的功能。但在实际使用中，复杂的版本控制功能并没有被过多地使用，在实际的内容管理中反而比较容易产生混乱，因此很多成

熟的 CMS 系统都没有引入版本控制的部分。

- 内容审批发布：内容经审批是为了方便编辑部门日常文章的发布和管理的功能。其具体的内容是，编辑发布的文章和消息需要经过管理员的审核后，才可以将内容发布的到网站上，技术上通常的设计是通过改变文章的标示位的参数来实现的。
- 内容采集：网站的内容来源通常由两个部分组成，就是原创内容（由编辑写稿发布）和采集内容（爬虫程序抓取）。使用内容采集功能，需要编辑人员定义采集规则，这些规则规定了需要采集内容的具体要求。通过采集的内容可以极大地丰富站点信息，因此内容采集功能也很受网站管理者和编辑人员的欢迎。
- 回收站功能：内容的修改操作中，由于数量巨大难免会出现误操作的情况，这时就需要通过回收站的功能将误删的内容找回来。在技术实现上，也是通过改变文章的标示位来实现的（编辑的删除操作只是改变了文章的标示状态，并没有真正的在数据库中删除对应的记录，只有在回收站清除该内容，才会在数据库中删除该条目）。
- 内容聚合：目前很多新闻站点都提供定制内容，即内容聚合（RSS）。内容聚合功能可以实时地获取定制的内容。
- 多国语言支持：既然 CMS 提供的在线翻译根本不可靠，就老老实实自己做多国语言内容好了。好的 CMS 允许创作者只提供某项内容的不同语言版本，由系统自动对这些不同语言版本的内容进行组织。

注意：支持多国语言的基础是对 UTF-8 的编码支持，CMS 的设计和程序代码需要使用 UTF-8 编码，为以后的功能插件扩展奠定编码的基础。

17.2　数据库设计

CMS 是对内容进行管理的后台系统，因此数据库的设计目的是按照栏目为每个管理栏目创建对应的表，用来存储内容。CMS 内容管理系统的数据库中用到了 4 张表，分别是文章内容（articles）表、注册会员（users）表、管理员账户（master）表和网站相关信息（sites）表。

CMS 内容管理系统数据库表含义如下。

- 文章内容（articles）表：存储与文章相关的所有信息。
- 注册会员（users）表：存储注册会员的个人资料和记录登录信息。
- 管理员账户（master）表：存储管理员账户名和密码。
- 网站相关信息（sites）表：存储网站辅助信息资料。

通过上面功能描述，可以清楚地了解到各个表的设计目的和需要完成的功能。下面来看具体的设计步骤。创建名为 cms 的数据库，SQL 语句如下：

```
CREATE DATABASE 'cms';
```

17.2.1　文章内容表

在 cms 数据库中，建立一个命名为 artcles 的文章内容表，结构如表 17.1 所示。创建表

及相关字段的 SQL 语句如下：

```
SET SQL MODE="NO AUTO VALUE ON ZERO";
--
-- 数据库: 'cms'
--
-- --------------------------------------------------------
--
-- 表的结构 'articles'
--
CREATE TABLE 'articles' (
  'ID' int(10) unsigned NOT NULL auto increment,
  'Title' varchar(200) NOT NULL,
  'Click' int(11) NOT NULL,
  'Content' longtext,
  'Date' date NOT NULL,
  'Category' smallint(6) NOT NULL,
  'VIP' tinyint(1) NOT NULL default '0',
  PRIMARY KEY  ('ID'),
  KEY 'Category' ('Category')
) ENGINE=MyISAM  DEFAULT CHARSET=gbk AUTO_INCREMENT=7277 ;
```

表 17.1　文章内容表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
ID	int	10	Not null	auto_increment	文章编号（自增主键）
Title	varchar	200	Not null		文章标题
Click	int	11	Not null		文章点击数
Content	longtext		Not null		文章内容
Date	date		Not null		文章创建日期
Category	smallint	6	Not null		文章所属分类
VIP	tinyint	1	Not null	0	是否为只有 VIP 可读

注意：文章内容表是一个基础的存储文章相关信息的表。读者可以根据实际的需要扩展该表，比如增加该文章的责任编辑、是否推荐到首页等功能。在扩展方式上，可以通过增加附表的方式实现。

为了便于后面程序的调试，向数据库中插入两条测试数据。插入测试代码的 SQL 语句如下：

```
--
-- 导出表中的数据 'articles'
--
INSERT INTO 'articles' VALUES (251, '康斯坦西奥：第二季度欧洲经济增长将减速', 82,
'<br />', '2008-05-27', 3, 0);
INSERT INTO 'articles' VALUES (252, '康斯坦西奥：必须等待 4 月和 5 月数据来评估经
济表现', 117, '<br />', '2008-05-27', 3, 0);
```

17.2.2　注册会员表

在 cms 数据库中，建立一个命名为 users 的注册会员表，结构如表 17.2 所示。创建表

及相关字段的 SQL 语句如下：

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
--
-- 数据库: 'cms'
--
-- --------------------------------------------------------
--
-- 表的结构 'users'
--
CREATE TABLE 'users' (
  'user_id' int(10) NOT NULL auto_increment,
  'username' varchar(255) NOT NULL default '',
  'user_regdate' datetime NOT NULL default '0000-00-00 00:00:00',
  'user_password' varchar(32) NOT NULL default '',
  'user_email' varchar(50) NOT NULL default '',
  'user_qq' varchar(50) NOT NULL default '',
  'user_website' varchar(100) NOT NULL default '',
  'user_viewemail' tinyint(1) NOT NULL default '0',
  'activity' int(1) NOT NULL default '0',
  PRIMARY KEY  ('user_id'),
  KEY 'username' ('username')
) ENGINE=MyISAM  DEFAULT CHARSET=gbk AUTO_INCREMENT=3 ;
```

表 17.2　注册会员表

字　段	数 据 类 型	长度	NULL	默　认　值	字 段 说 明
user_id	int	10	Not null	auto_increment	注册用户 id（自增主键）
username	varchar	255	Not null		注册用户名
user_regdate	datetime		Not null	0000-00-00 00:00:00	注册时间
user_password	varchar	32	Not null		登录密码（MD5 加密串）
user_email	varchar	50	Not null		接受密码邮件地址
user_qq	varchar	50	Not null		会员 QQ
user_website	varchar	100	Not null		企业网站
user_viewemail	tinyint	1	Not null		是否显示邮件地址
activity	int	1	Not null	0	账户状态 1 为启用，0 为停用

注册会员表是用来存储注册会员相关信息的表。在实际的应用中，由于对注册会员的信息要求不同，所以在字段设置上会有所不同。读者可以根据实际情况扩展该表，在扩展方式上可以通过增加附表的方式实现。

注意：注册会员登录密码字段存储的加密后的字符串不要直接使用明文，以免程序异常时暴露登录密码。

同样地，为了便于后面程序的调试，向数据库中插入两条测试数据。插入测试代码的 SQL 语句如下：

```
INSERT INTO 'users' VALUES (1, 'admin', '2009-04-07 21:00:14', '', 'dsd',
'', '', 0, 1);
INSERT INTO 'users' VALUES (2, 'test', '2009-04-07 21:48:21', '', '', '222',
0, 1);
```

17.2.3　管理员账户表

在 cms 数据库中，建立一个命名为 master 的管理员账户表，结构如表 17.3 所示。创建表及相关字段的 SQL 语句如下：

```
SET SQL MODE="NO AUTO VALUE ON ZERO";

--
-- 数据库: 'cnau95 com'
--
-- --------------------------------------------------------
--
-- 表的结构 'master'
--

CREATE TABLE 'master' (
  'ID' int(11) NOT NULL auto increment,
  'UserName' varchar(50) default NULL,
  'Password' varchar(50) default NULL,
  PRIMARY KEY  ('ID')
) ENGINE=MyISAM  DEFAULT CHARSET=gbk AUTO_INCREMENT=2 ;
```

表 17.3　管理员账户表

字　段	数 据 类 型	长度	NULL	默　认　值	字 段 说 明
ID	int	11	Not null	auto_increment	管理员 id（自增主键）
UserName	varchar	50	Not null		登录用户名
Password	varchar	50	Not null		登录密码 MD5 加密串

管理员账户表是用来存储管理员账户名和密码的表，该表存储管理员密码为 MD5 加密后的字符串。

同样地，为了便于后面程序的调试，向数据库中插入一条测试数据。插入测试代码的 SQL 语句如下：

```
INSERT INTO 'master' VALUES (1, 'admin', '21232f297a57a5a743894a0e4a801fc3');
```

17.2.4　网站相关信息表

在 cms 数据库中，建立一个命名为 sites 的网站相关信息表，结构如表 17.4 所示。创建表及相关字段的 SQL 语句如下：

```
SET SQL MODE="NO AUTO VALUE ON ZERO";

--
-- 数据库: 'cms'
--
-- --------------------------------------------------------
--
-- 表的结构 'sites'
--
CREATE TABLE 'sites'  (
```

```
  'Content' mediumtext NOT NULL,
  'Title' varchar(50) NOT NULL,
  UNIQUE KEY 'Category 2' ('Title'),
  KEY 'Category' ('Title')
) ENGINE=MyISAM DEFAULT CHARSET=gbk;
```

表 17.4　网站相关信息表

字　段	数 据 类 型	长度	NULL	默 认 值	字 段 说 明
Content	mediumtext		Not null		网站相关信息的内容
Title	varchar	50	Not null		栏目名称

网站相关信息表是用来存储网站相关栏目内容的表，每一个栏目名称后面对应一个具体的相关信息的内容。

同样地，为了便于后面程序的调试，向数据库中插入一条测试数据，插入测试代码的 SQL 语句如下：

```
--
-- 导出表中的数据 'sites'
--
INSERT INTO 'sites' VALUES ('<P><FONT style="FONT-SIZE: 20px">网站相关栏目
</FONT></P>\r\n<P> </P>\r\n<P>关于我们介绍内容</P>', '关于我们');
```

17.3　管理框架搭建

CMS 最直观的表现就是一个管理的后台，使用者可以通过它直观地管理网站的信息。本章将介绍如何搭建一个管理的框架显示基本的系统信息，以及创建搭建管理功能的基础类库。

17.3.1　创建管理框架

CMS 后台管理框架由 4 个部分组成，即登录验证程序（login.php）、导航栏部分（top.php）、侧边栏部分（left.php）、主体部分（main.php），它们共同构成了后台管理的框架界面。下面介绍各个部分的功能和在整个框架中的作用。

- 登录验证程序：验证管理员的身份信息，创建登录的状态的 Session 变量。
- 导航栏部分：负责显示功能管理模块的导航条（添加和删除管理栏目也可以对该文件进行编辑操作）。
- 侧边栏部分：显示对应模块下的栏目。
- 主体部分：根据传入值不同，载入并显示不同的操作程序界面。

现在逐步介绍框架搭建过程和程序代码的实现。创建入口文件（index.php），该文件并不需要处理具体的逻辑，而是通过<frame>标签将各个组成部分加载到框架中。入口文件的具体代码样式如下：

```
<title>后台管理</title>
<meta http-equiv="Content-Type" content="text/html; charset=gbk" />
```

```
</head>
<frameset rows="80,*" cols="*" frameborder="no" border="0"
    framespacing="0">
    <frame src="top.php" name="topFrame" scrolling="No" noresize="noresize"
        id="topFrame" title="topFrame" />
    <frameset cols="180,*" frameborder="no" border="0" framespacing="0">
        <frame src="left.php" name="leftFrame" scrolling="auto"
            noresize="noresize" id="leftFrame" title="leftFrame" />
        <frame src="main.php" name="mainFrame" scrolling="yes"
            noresize="noresize" id="mainFrame" title="mainFrame" />
    </frameset>
</frameset>
<noframes>
<body>
<p>Please Use IE 5.0</p>
```

17.3.2　登录验证程序

上述代码将之前介绍的组成管理框架的程序都加载到一个界面中。但是，由于管理后台的重要性，不可能将访问地址不做任何身份验证直接发布出去，所以创建身份认证的程序插入到框架程序之前，代码如下：

```
<?php
session_start();                //初始化一个新的 Session
include("check_user.php");
?>
```

【代码解读】

这里创建了一个身份验证程序 check_user.php，用对管理员进行身份验证。该文件代码如下：

```
<?php
if (!$_SESSION['admin'])        //判断是否为管理员
    header('location: login.php');
?>
```

【代码解读】

该文件向浏览器声明了一个 header()的头部跳转信息，该信息将使浏览器跳转到登录验证（login.php）程序。该文件的代码如下：

```
<?php
require_once '../init.inc.php';     //加载配置类文件
//判断操作类型
if ($_GET ['ac'] == 'logout') {
    session_destroy ();     //结束目前的 Session（没有参数，且返回值均为 true）
}
usdb ( 'master' );          //选择指定的数据库
if ($_POST ['UserName']) {
    if ($ID = $master->getID ( trim ( $_POST ['UserName'] ) )) {
        $row = $master->getVar ( $ID );                     //获得 id
        if (md5 ( $_POST ['Password'] ) == $row ['Password'
                            //MD5 对表单传递过来的密码与数据中密码比对
{
            $_SESSION ['admin'] = $_POST ['UserName'];
```

```
            $_SESSION ['admin_id'] = $ID;
            header ( 'location: index.php' );          //浏览器重定向
        } else
            $msg = '密码错误';
    } else
        $msg = '无此用户';
}
?>
```

上述程序对应的模板部分包含表单验证程序和页面模板程序。表单验证程序的核心代码如下：

```
<title>登录管理系统</title>
<meta http-equiv="Content-Type" content="text/html; charset=gbk" />
<script>
<?php
if ($msg)                                  //判断变量是否存在
    echo "alert('$msg');";                 //抛出报错信息
?>
<!--表单检查函数-->
function check_form()
{
    var o= document.forms[0];
    <!--判断用户名是否为空-->
    if(o.UserName.value=='')
    {
        alert('请填写登录名');
        o.UserName.focus();
        return false;
    }
    <!--判断密码是否为空-->
    if(o.Password.value=='')
    {
        alert('请填写密码');
        o.Password.focus();
        return false;
    }
    return true;
}
</script>
</head>
```

【代码解读】

以上程序定义了表单函数 check_form()，该函数的功能是检查用户名是否为空、检查输入密码是否为空，如果异常抛出报错信息。

页面模板程序代码部分的核心代码如下：

```
<form action="login.php" method="post" target="_parent" name="login"
    onsubmit="return check_form();">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td colspan="2">
        <table width="100%" border="0" cellspacing="0" cellpadding="0">
            <tr>
                <td width=2><img src="images/line_lft.png"></td>
                <td background="images/loginline.png"
                    style="color: white; font-weight: bold; height: 30px;
```

```
                    padding-left: 10px;">Administrator
                Login</td>
                <td><img src="images/line_rt.png"></td>
            </tr>
        </table>
        </td>
    </tr>
    <tr>
        <td colspan="2"> </td>
    </tr>
    <tr class="tr">
        <td width="30%" style="padding-left: 15px;">用户名:</td>
        <td width="70%"><input name="UserName" type="text" class="input" />
        </td>
    </tr>
    <tr class="tr">
        <td style="padding-left: 15px">密   码:</td>
        <td><input name="Password" type="password" class="input" /></td>
    </tr>
    <tr class="tr">
        <td colspan="2" align="center"><input type="submit" name="Submit"
            value="登录" class="btn" /> <input type="reset" name="Submit2"
            value="重置" class="btn" /></td>
    </tr>
    <tr class="tr">
        <td colspan="2" align="center"> </td>
    </tr>
</table>
</form>
```

【代码解读】

登录验证程序首先判断表单传递过来的操作类型，因为管理员登出的程序也是该程序。操作类型判断的代码如下。

```
//判断操作类型
if($_GET['ac'] == 'logout'){
    session_destroy();      //销毁 Session
}
```

程序执行的步骤如下。

（1）当管理员选择登出后会将 ac 变量赋值为 logout，程序判断了操作类型后就会销毁创建的用于记录身份信息的 session 变量。为了便于程序加载不同的操作类，创建一个公共配置文件（init.inc.php），该文件包含了各类操作的基类。下面的代码的作用是加载管理员账户（master）类，这个类的构成将在后面基础类库中做详细说明。

```
//管理员账户(master)类:
usdb('master');             //加载指定的数据库
```

（2）用管理员账户（master）类对表单提交的登录信息进行验证。判断及验证代码如下：

```
if($ID=$master->getID(trim($_POST['UserName'])))
    {
        $row =$master->getVar($ID);                          //获得管理员 id
        if (md5($_POST['Password']) == $row['Password'])
                               //MD5 对表单传递过来的密码与数据中密码比对
```

```
        {
            $_SESSION['admin'] = $_POST['UserName'];
            $_SESSION['admin_id'] = $ID;
            header('location: index.php');      //header()函数重定向浏览器
        }else
            $msg = '密码错误';
    }else
        $msg = '无此用户';
}
```

在浏览器中执行下面的链接：

```
http://localhost/cms/admin/login.php
```

CMS 系统登录界面程序的浏览效果如图 17.1 所示。



图 17.1　CMS 系统登录界面

在输入框中输出测试数据验证程序执行流程。当输入正确的管理员名称和错误的密码时，页面显示如图 17.2 所示，弹出错误提示对话框并终止下面的执行。

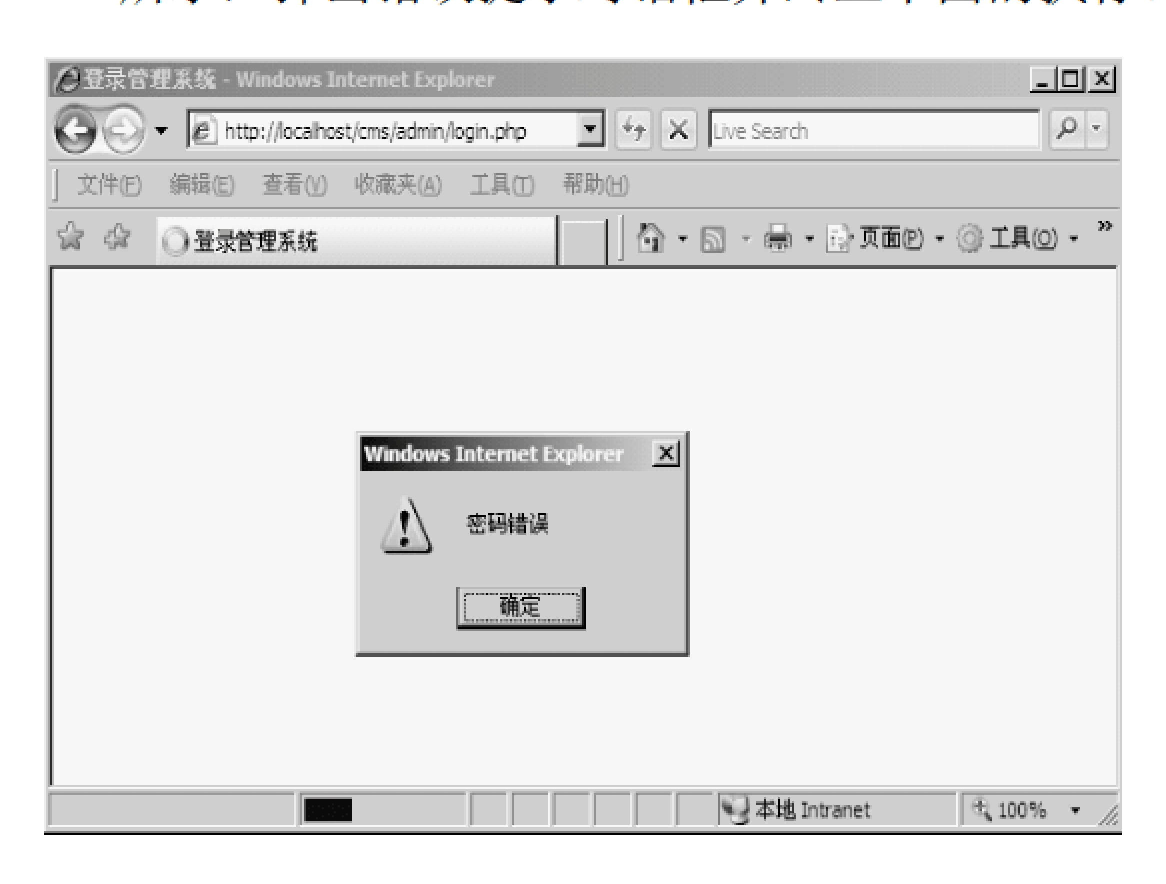


图 17.2　登录异常验证

17.3.3　导航栏部分

导航栏程序部分主要是通过手动修改页面的方式，将栏目信息写入到页面代码中实现

的，请看具体的代码：

```
<div class="frameTop">
    <div class="nav">
        <a href="../" title="站点首页" target="_blank">站点首页</a>
        <a href="login.php?ac=logout" title="Login out" target="_parent"
        onClick="return loginOut();">退出登录</a>
        </div>
    <div class="topLogo"></div>
</div>
<div class="navPanel">
    <div class="innerNavPanel">
        <div class="topButtons">
        <a href="./" target="_top">管理首页</a>
        <a href="left.php?m=site" target="leftFrame">网站相关</a>
        <a href="left.php?m=artiles" target="leftFrame">文章管理</a>
        <a href="left.php?m=user" target="leftFrame">会员管理</a>
        <a href="left.php?m=1" target="leftFrame">待添加栏目一</a>
        <a href="left.php?m=2" target="leftFrame">待添加栏目二</a>
        <a href="left.php?m=3" target="leftFrame">待添加栏目三</a>
        </div>
    </div>
</div>
<script language="javascript">
function loginOut(){
    var msg = confirm("确定要退出系统么？");
    if(!msg){
        return false;
    }
}
</script>
```

【代码解读】

管理员登出的部分放入到导航栏代码中，根据登录验证程序中规定的操作的判断，将 ac 变量赋值为 logout，并写入一段 JavaScript 代码用来提示用户是否退出系统，防止管理员误操作。上面的代码的核心部分是规定侧边栏（left.php）的传入参数的形式，链接形式如下：

```
<a href="left.php?m=规定参数" target="leftFrame">栏目名称</a>
```

在浏览器中执行下面的链接：

```
http://localhost/cms/admin/top.php
```

CMS 系统导航栏界面程序的浏览效果，如图 17.3 所示。

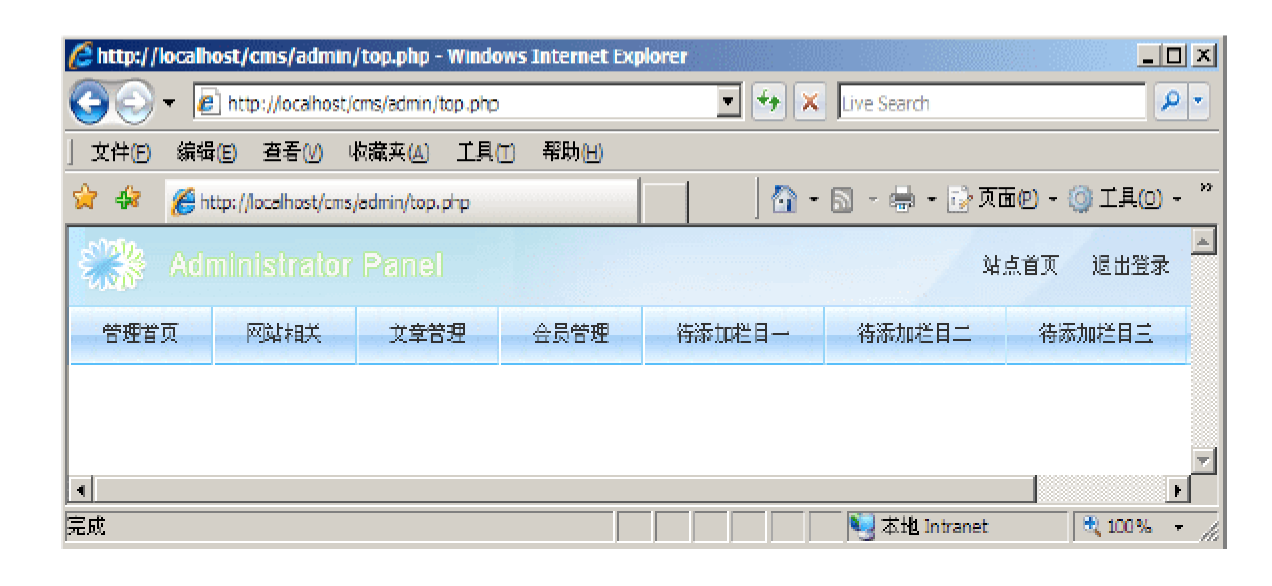


图 17.3　导航栏界面

17.3.4　侧边栏部分

侧边栏导航的功能是显示导航链接，实现的原理是通过链接传入的参数值，选择需要显示管理模块代码。现在以“网站相关”栏目为例子介绍侧边栏显示代码的格式。“网站相关”栏目的页面代码如下：

```
<div class="t" style="display: none" id="site">
<table border="0" cellspacing="0" cellpadding="0" width=100%>
    <tr class="leftH">
        <td>网站相关</td>
    </tr>
    <tr class="leftmenu" onmouseover="changeBg(this,1);"
        onmouseout="changeBg(this,0);">
        <td><a href="site.php?t=关于我们" target="mainFrame">关于我们</a>
        </td>
    </tr>
    <tr class="leftmenu" onmouseover="changeBg(this,1);"
        onmouseout="changeBg(this,0);">
        <td><a href="service.php" target="mainFrame">联系方式</a></td>
    </tr>
    <tr class="leftmenu" onmouseover="changeBg(this,1);"
        onmouseout="changeBg(this,0);">
        <td><a href="links.php" target="mainFrame">友情连接</a></td>
    </tr>
    <tr class="leftmenu" onmouseover="changeBg(this,1);"
        onmouseout="changeBg(this,0);">
        <td><a href="links_save.php" target="mainFrame">新增连接</a></td>
    </tr>
    <tr class="leftmenu" onmouseover="changeBg(this,1);"
        onmouseout="changeBg(this,0);">
        <td><a href="pwd.php" target="mainFrame">修改密码</a></td>
    </tr>
    <tr class="leftmenu" onmouseover="changeBg(this,1);"
        onmouseout="changeBg(this,0);">
        <td><a href="login.php?ac=logout" target="_top">退出登录</a></td>
    </tr>
</table>
</div>
```

【代码解读】

通过上面的代码可以了解到，整个侧边栏是通过以下的模型叠加在一起的：

```
<div class="t" style="display:none" id="操作参数">
<table border="0" cellspacing="0" cellpadding="0" width=100%>
<tr>
<td>内容部分</td>
</tr>
</table>
```

下面创建一个 JavaScript 脚本，该脚本的作用是根据传入的参数值设置<div>标签的 style 参数，当 style.display = 'block';时表示显示该标签中的内容，其他部分的标签的内容部分将会被隐藏。该脚本的代码如下：

```
<script language="javascript">
```

```
<!--根据参数值选择不同菜单-->
function changeBg(obj,value){
    if(value==1)
        var c = 'leftmenu2';
    else
        var c = 'leftmenu';
    obj.className=c;
}
<?php
$m = 'default';              //设置默认状态
if($_GET['m'])
    $m = $_GET['m'];         //获得当前状态
?>
document.getElementById('<?php echo $m?>').style.display = 'block';
</script>
```

在浏览器中执行下面的链接：

```
http://localhost/cms/admin/left.php?m=site
```

CMS 系统“网站相关”栏目侧边栏程序的浏览效果，如图 17.4 所示。

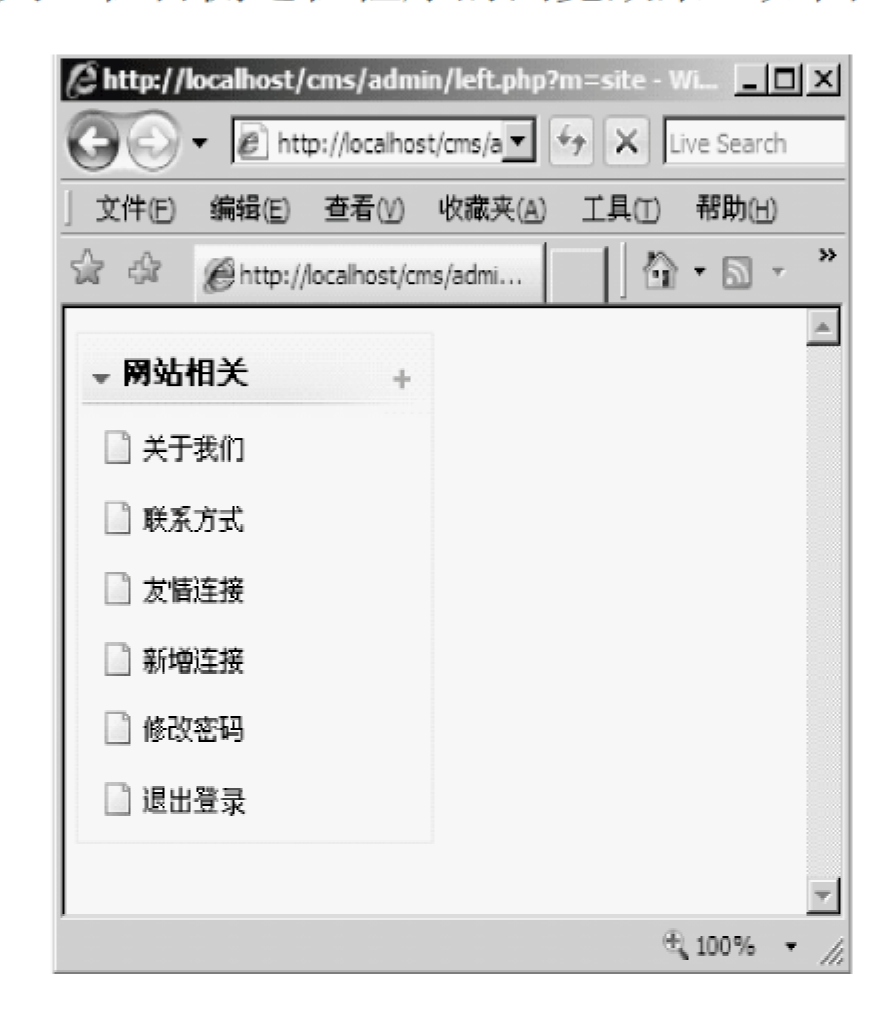


图 17.4 “网站相关”栏目侧边栏

17.3.5　主体部分

框架的内容主体部分默认显示系统相关的信息，便于管理员直观地获得系统的基本信息和资源使用情况。CMS 管理框架的默认显示主体程序（main.php）代码如下：

```
<?php
include('../init.inc.php');     //加载配置文件
?>
<!--主体框架模板-->
<base target="mainFrame">
<div class="m"></div>
<div class="t">
<table align=center cellspacing=0 cellpadding=0>
    <tr class=head>
        <td colspan="2">系统信息</td>
```

```
    </tr>
    <tr class=line>
        <td>程序路径：</td>
        <td><?php echo ROOT PATH?></td>
    </tr>
    <tr class=line>
        <td width="20%">PHP 版本：</td>
        <td><?php echo PHP_VERSION?></td>
    </tr>
<!--数据库使用信息模板-->
    <tr class=line>
        <td>MySQL 版本：</td>
        <td>
<?php
    //获得数据库当前使用情况
    $row = $db->get row("SELECT VERSION() AS dbversion",ARRAY A);
    echo $row['dbversion'];                //输出数据信息
?></td>
    </tr>
    <tr class=line>
        <td>Web 服务器：</td>
        <td>
    <?php
        echo $ SERVER['SERVER SOFTWARE'];  //服务器信息
    ?>
    </td>
    </tr>
    <tr class=line>
        <td>远程文件获取：</td>
        <td>
    <?php
    //获得远程文件信息
    echo ini get('allow url fopen') ? '<span style="color:green"> Supported
    </span>' : '<span style="color:red">Not supported</span>';
    ?>
        </td>
    </tr>
    <tr class=line>
        <td>最大上传限制：</td>
        <td>
            <?php
                //获得文件最大上传限制
                echo ini get('file uploads') ? ini get('upload max  filesize')
                : '<span style="color:red">Disabled</span>';
            ?></td>
    </tr>
    <tr class=line>
        <td>服务器时间：</td>
        <td><?php echo date('Y-m-d H:i:s')?></td>
    </tr>
<?php
    //获得数据表情况的数组
    $rows = $db->get results("SHOW TABLE STATUS",ARRAY A);
    foreach ($rows as $row) {
        $o_size = $o_size + $row['Data_length'] + $row['Index_length'];
    }
    $o_size= number_format($o_size/(1024*1024),2);
?>
    <tr class=line>
```

```
        <td>数据库占用：</td>
        <td><?php echo $o size?> M</td>
    </tr>
    <tr class=line>
        <td> </td>
        <td> </td>
    </tr>
</table>
</div>
```

【代码解读】

上面的代码通过 PHP 内置的函数获取系统和版本的信息，这些信息是系统运行的基础信息，但是常常被开发者所忽略。这部分代码的获取方法都是固定的，读者可以将这部分代码保存为功能模块，以便于在今后的开发中直接使用。

在浏览器中执行下面的链接：

```
http://localhost/cms/admin/main.php
```

CMS 系统框架的内容主体部分默认显示系统相关的信息。浏览效果如图 17.5 所示。

图 17.5　默认系统信息

17.4　基础类库介绍

类是变量与作用于这些变量的函数的集合，被广泛地应用在面向对象设计编程（oop）的开发模式中。在实际的开发中，类通常被用来描述某一类事务，它是针对某一类业务的操作集合。在本章 CMS 模块中就使用了创建基础类库的方式搭建平台。下面根据业务的需求具体介绍 CMS 中核心的几个基础的类，它们包含了对数据库的基本操作和对具体业务模型的基本操作。后面在此之上搭建的程序就必须考虑具体的操作过程了。面向对象设计编程（OOP）极大地提高了系统的开发效率，读者在学习这部分的时候可以仔细体会。

17.4.1　公共类

公共类程序（common.php）是其他针对数据库操作类的公共类，它定义了针对数据库操作的几个公共的变量。下面请看公共类程序的具体代码：

```
<?php
class Common
{
    var $db;
    var $recordNum;
    //公共方法
    function Common()
    {
        global $db;                     //创建全局变量（数据库链接）
        if (!isset($db))                //检查 db 异常
            return false;
        $this->db = $db;                //实例化数据库
    }
    //异常排错方法
    function debug()
    {
        $this->db->debug();             //报错抛出
    }
}
?>
```

【代码解读】

上面的代码定义了两个方法：

（1）创建对数据库操作的句柄。

（2）程序异常时的报错抛出方法。

17.4.2　文章操作类

文章操作类继承了公共类，是封装对数据库中 articles 表常用操作的类，类中定义的方法如下。

- ❑ Add()方法：增加文章（insert）操作。如果成功，返回文章 id；插入失败，返回报错。
- ❑ Update()方法：更新指定 id 的文章（update）操作。
- ❑ getVar()方法：获得指定 id 的文章内容操作。如果成功，返回获得内容数组。
- ❑ getList()方法：获得全部文章列表操作。如果成功，返回获得内容数组。
- ❑ setClick()方法：更新指定文章的点击数。无返回值。
- ❑ Del()方法：删除指定的文章操作。

了解了文章操作类中方法的具体作用后，创建文章操作类程序（articles.php）。文件中具体代码如下：

```
<?php
require_once ('common.php');            //加载公共文件
class Articles extends Common {
```

```
    var $tableName = "articles";        //表名变量
    function Articles() {
        Common::Common ();
    }
    //添加文章方法
    function add($vars) {
        $sql = "INSERT INTO
                " . $this->tableName . "
                (Title,Content,Click,Category,'Date')
                VALUES
                ('$vars[Title]','$vars[Content]',0,'$vars[Category]','" .
                date ( 'Y-m-d' ) . "')";
        if ($this->db->query ( $sql ))
            return $this->db->insert id;
        return false;
    }
    //更新文章方法
    function update($vars) {
        $sql = "UPDATE " . $this->tableName . " SET
        'Title'='$vars[Title]', 'Content'='$vars[Content]'
        WHERE ID=$vars[ID]";
        return $this->db->query ( $sql );
    }
    //获得指定 id 的文章内容
    function getVar($ID) {
        return $this->db->get_row ( "SELECT * FROM " . $this->tableName .
        " WHERE ID=$ID", ARRAY A );
    }
    //获得全部文章列表
    function getList($str = NULL, $page size = 0, $page = 0) {
        $this->recordNum = $this->db->get var ( "SELECT COUNT(ID) AS n FROM
        " . $this->tableName . " WHERE 1 $str" );
        if (1 > $this->recordNum)
            return NULL;
        $sql = "SELECT * FROM " . $this->tableName . " WHERE 1 $str ";
        if (1 < $page size && 0 < $page)
            $sql .= 'LIMIT ' . ($page - 1) * $page_size . ',' . $page_size;
        return $this->db->get results ( $sql, ARRAY A );
    }
    //更新点击数
    function setClick($ID) {
        $sql = "UPDATE " . $this->tableName . " SET
        Click=Click+1
        WHERE ID=$ID";
        return $this->db->query ( $sql );
    }
    //删除指定的文章
    function del($ID) {
        $sql = "DELETE FROM " . $this->tableName . " WHERE ID=$ID";
        $this->db->query ( $sql );
        return true;
    }
}
?>
```

【代码解读】

上面的 articles 类是完成了对数据库中 articles 表的基本操作，包括插入、更新、删除、查询等常用操作。

17.4.3　管理员账户操作类

管理员账户操作类继承了公共类，是封装对数据库中 master 表常用操作的类，类中定义的方法如下。

- Add()方法：增加一个管理员账户（向 master 表中插入一条数据）。创建成功返回 true 标识，失败则返回 false 标识。
- Update()方法：更新指定 id 的管理员的信息（用户名、密码）。
- getVar()方法：获得指定 id 的管理员信息。
- getID()方法：根据用户名获得用户 id。
- getList()方法：获得全部的管理员信息列表。
- Del()方法：删除指定的管理员信息，删除后返回 true 标识。

管理员账户操作类的具体代码如下：

```
<?php
require_once ('common.php');          //加载公共文件
class Master extends Common {         //继承基类（管理员账户操作类）
    var $tableName = "master";
    function Master() {
        Common::Common ();
    }
    //增加一个管理员账户
    function add($vars) {
        $sql = "INSERT INTO
                " . $this->tableName . "
                (UserName, 'Password')
                VALUES
                ('$vars[UserName]','$vars[Password]')";
        //die($sql);
        if ($this->db->query ( $sql ))
            return true;
        return false;
    }
    //更新管理员的信息(用户名、密码)
    function update($vars) {
        $sql = "UPDATE " . $this->tableName . " SET UserName='$vars[UserName]',
        'Password'='$vars[Password]' WHERE ID=$vars[ID]";
        return $this->db->query ( $sql );
    }
    //获得指定 id 的管理员信息
    function getVar($ID) {
        return $this->db->get_row ( "SELECT * FROM " . $this->tableName .
        " WHERE ID=$ID", ARRAY_A );
    }
    //根据用户名获得用户 id
    function getID($Title) {
        return $this->db->get_var ( "SELECT * FROM " . $this->tableName .
        " WHERE 'UserName'='$Title'" );
    }
    //获得全部的管理员列表
    function getList($str) {
        $sql = "SELECT * FROM " . $this->tableName . " WHERE 1 $str ORDER
        BY OID DESC";
```

```
        return $this->db->get results ( $sql, ARRAY A );
    }
    //删除指定的管理员
    function del($ID) {
        $sql = "DELETE FROM " . $this->tableName . " WHERE ID=$ID";
        $this->db->query ( $sql );
        return true;
    }
}
?>
```

【代码解读】

上面代码中，封承了公共类，是封装对数据库中 master 表常用操作的类，该类中定义了针对管理员账户的常用操作（参看该类定义方法说明）。

17.4.4 注册会员操作类

注册会员操作类继承了公共类，是封装对数据库中 users 表常用操作的类，类中定义的方法如下。

- Add()方法：增加一个新注册用户，首先检查用户名是否存在，然后向 users 表中插入一条新的注册数据。
- Update()方法：更新指定的已注册用户信息，即注册时的全部注册资料信息。
- getVar()方法：获得指定 user_id 对应的信息。
- getList()方法：获得用户信息列表，该方法包含分页信息的设置（页数、起始值）等信息。
- Del()方法：删除指定用户的信息。
- update_status()方法：更新指定用户的账户状态。
- set_password()方法：重置注册用户密码。

注册会员操作类的代码如下：

```
<?php
require_once ('common.php');        //加载公共文件
class Users extends Common          //注册会员操作类继承公共类
{
    var $tableName = "users";       //操作数据库表名
    var $activity = 1;
    function Users() {
        Common::Common ();
    }
    //增加一个新注册用户
    function add($vars) {
        $row = $this->db->get_row ( "SELECT * FROM " . $this->tableName .
        "WHERE username='$vars[username]'", ARRAY A );
        if ($row ['username']) {
            return false;
        }
        $sql = "INSERT INTO
                " . $this->tableName . "
(username,user regdate,user password,user email,user qq,user website,user
viewemail,activity)
                VALUES
```

```
                ('$vars[username]','" . date ( 'Y-m-d H:i:s' ) . "','" . md5
                ( $vars ['user password'] ) . "','$vars[user email]',
                '$vars[user qq]','$vars[user website]',
                '$vars[user_viewemail]',{$this->activity})";
        $this->db->query ( $sql );
        $thisID = $this->db->insert id;
        return $thisID;
    }
    //更新已注册用户信息
    function update($vars) {
        $sql = "UPDATE " . $this->tableName . " SET
                'user email'='$vars[user email]', 'user qq'='$vars
                [user-99]',
                'user website'='$vars[user website]', 'user viewemail'='
                $vars[user viewemail]'
                WHERE user id=$vars[user id]";
        return $this->db->query ( $sql );
    }
    //获得指定 user_id 对应的信息
    function getVar($ID) {
        return $this->db->get_row ( "SELECT * FROM " . $this->tableName .
        "WHERE user id=$ID", ARRAY A );
    }
    //获得用户信息列表
    function getList($str = NULL, $page size = 0, $page = 0) {
        $this->recordNum = $this->db->get_var ( "SELECT COUNT(*) AS n FROM".
        $this->tableName . " WHERE 1 $str" );
        if (1 > $this->recordNum)
            return NULL;
        $sql = "SELECT * FROM " . $this->tableName . " WHERE 1 $str ";
        if (1 < $page size && 0 < $page)
            $sql .= 'LIMIT ' . ($page - 1) * $page_size . ',' . $page_size;

        //die($sql);
        return $this->db->get results ( $sql, ARRAY A );
    }
    //删除指定用户的信息
    function del($ID) {
        $sql = "DELETE FROM " . $this->tableName . " WHERE user_id=$ID";
        $this->
        db->query ( $sql );
        return true;
    }
    //更新指定用户的账户状态
    function update status($user id, $activity) {
        $this->db->query ( "update {$this->tableName} set activity=
        {$activity} where user id={$user id}" );
    }
    //重置注册用户密码
    function set password($user id, $password) {
        $this->db->query ( "update {$this->tableName} set user password='" .
        md5 ( $password ) . "' where user id={$user id}" );
    }
}
?>
```

注意：set_password 方法重置的密码是 md5 加密后的串。

17.5　后台管理功能

经过前面的介绍，读者已经了解了如何搭建后台管理平台的框架，并且也已经创建了对业务相关的数据库基础的类。其中封装了基础的操作，这极大地提高了搭建后续程序的效率。本节将应用这些基础的类，继续实现后台管理功能的基本功能。

17.5.1　会员管理

会员管理是 CMS 系统中使用频率较高的一个管理功能。因为通过会员管理可以直接反应出一些网站的运营情况，如广告投放效果的检测等。但是，有时不可避免的会出现一些“休眠”或者是“垃圾”注册用户，它们大量地占据了数据库的资源。因此，对会员进行定时的清理就十分必要，同时对一些出现异常的账户停止会员功能。

下面具体介绍会员后台管理功能的代码实现。会员管理程序（users.php）的代码如下：

```
<?php
include('../init.inc.php');
usdb("users");                                      //加载用户数据库类
uslib("page");                                      //加载分页相关类
$page = $_GET['page'] ? $_GET['page'] : 1;          //获得当前页码，默认为第 1 页
$page_size = 12;                                    //偏移量（行数）
if($_GET['del']){                                   //删除操作
    $users->del($_GET['del']);
    header("location:users.php?page=$page");        //重定向浏览器
}
//更新账户状态
if($_GET['doid']){
    $users->update_status($_GET['doid'],$ac);
    header("location:users.php?page=$page");        //重定向浏览器
}
$rows = $users->getList($strWhere,$page_size,$page);  //获得用户数组
$pager = new listPage($users->recordNum,$page_size,8);
$pager->lastPage = '<font face="webdings">7</font>';
$pager->nextPage = '<font face="webdings">8</font>';
$pager->firstPage = '<font face="webdings">9</font>';
$pager->endPage = '<font face="webdings">:</font>';
$pager->cssClass = '';
$pager->createPage();                               //实例化翻页类
?>
<base target="mainFrame">
<div class="m"></div>
<div class="t">
<table align=center cellspacing=0 cellpadding=0>
<tr class=head>
  <td colspan="5">
用户管理</td>
</tr>
<tr class=line>
  <td><strong>登录名</strong></td>
  <td><strong>QQ</strong></td>
```

```
  <td><strong>注册日期</strong></td>
  <td><strong>账号状态</strong></td>
  <td><strong>删除</strong></td>
</tr>
<?php
foreach($rows as $row){                                    //循环用户管理列表
?>
<tr class=line>
  <td width="20%"><?php echo $row['username']?></td>
  <td><?php echo $row['user_qq']?> </td>
  <td><?php echo $row['user_regdate']?></td>
  <td>
  <?php
    if($row['activity'] == 1)                              //判断账户当前状态
        echo '<font color="blue">启用</font> | <a href="?page='.$page.'
        "&doid='.$row['user_id'].'&ac=0" style="color:red">停用</a>';
    else
        echo '<font color="red">停用</font> | <a href="?page='.$page. '
        "&doid='.$row['user_id'].'&ac=1" style="color:blue">启用</a>';

  ?></td>
  <td><input name="Submit2" type="button" class="btn" value="删除" onclick="
  if(confirm('确定删除? ')){window.location='?del=<?php echo
  $row['ID']?>&page=<?php echo $page?>';}" /></td>
</tr>
<?php
}
?>
<tr class=line>
  <td colspan="5" align="center"><?php echo $pager->pageAll;?></td>
</tr>
</table>
</div>
```

【代码解读】

由于使用注册会员操作类封装了常用的数据库操作，所以在业务处理部分代码非常简洁。根据业务的需求首先初始化注册会员操作类（users 类），然后设定基本分页参数，实现代码如下：

```
include('../init.inc.php');
usdb("users");                                      //注册会员操作类
uslib("page");                                      //分页类
$page = $_GET['page'] ? $_GET['page'] : 1;          //获得当前页面，默认为第一页
$page_size = 12;                                    //页面偏移量（行数）
```

【代码解读】

上面加载了配置文件，初始化分页类（page 类）和注册会员操作类（users 类），然后设定了会员列表从第一页开始，每页的跨度是 12 条记录，即分页偏移值。最后程序判断具体动作的类型来执行对应的数据库操作。这里有两个功能，即修改账户状态（启用、停用）和删除账户。代码如下：

```
//删除操作
if($_GET['del']){
    $users->del($_GET['del']);
    header("location:users.php?page=$page");        //浏览器重定向
```

```
}
//更新账户状态
if($_GET['doid']){
    $users->update_status($_GET['doid'],$ac);       //更新账户
    header("location:users.php?page=$page");        //浏览器重定向
}
```

在上面的代码中，当动作的类型是 del 时，执行调用 users 类中的 del 方法删除对应账户，具体语句如下：

```
$users->del($_GET['del']);
```

如果动作类型是 doid 时，执行调用 users 类中的 update_status()方法更新账户状态标识位，实现账户的启用、停用操作，具体的语句如下：

```
$users->update_status($_GET['doid'],$ac);
```

在浏览器中执行下面的链接：

```
http://localhost/cms/admin/user.php
```

CMS 后台会员管理的界面浏览效果如图 17.6 所示。

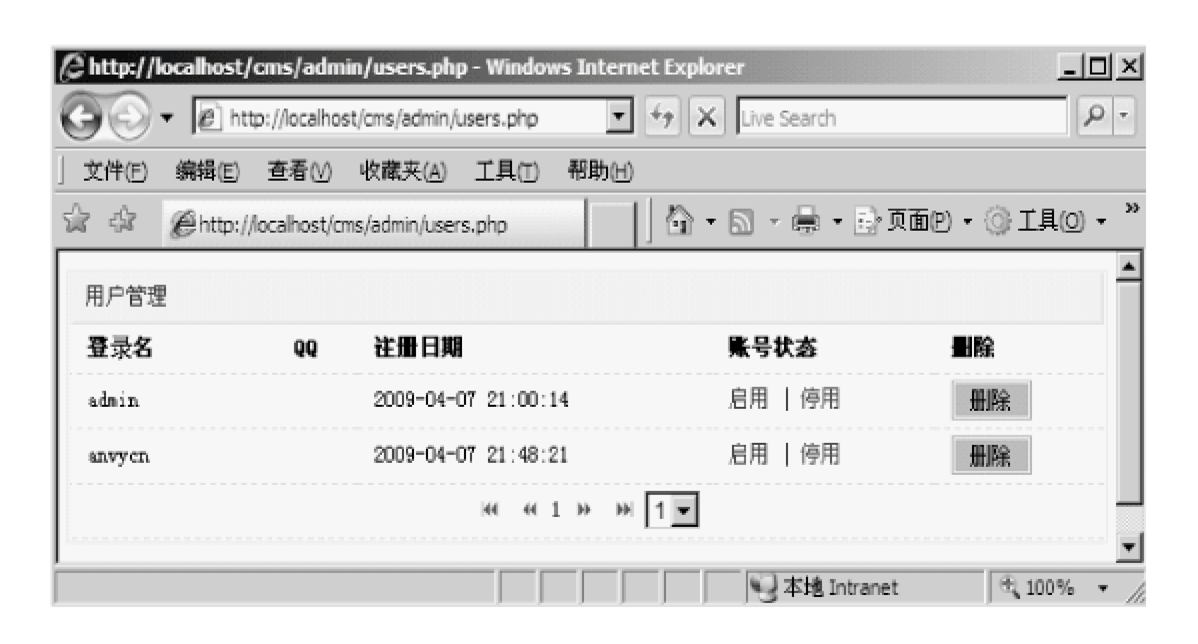


图 17.6　后台会员管理界面

选择一个用户，单击“删除”按钮后，删除提示页面如图 17.7 所示。单击“确认”按钮后完成了删除指定账户的操作。

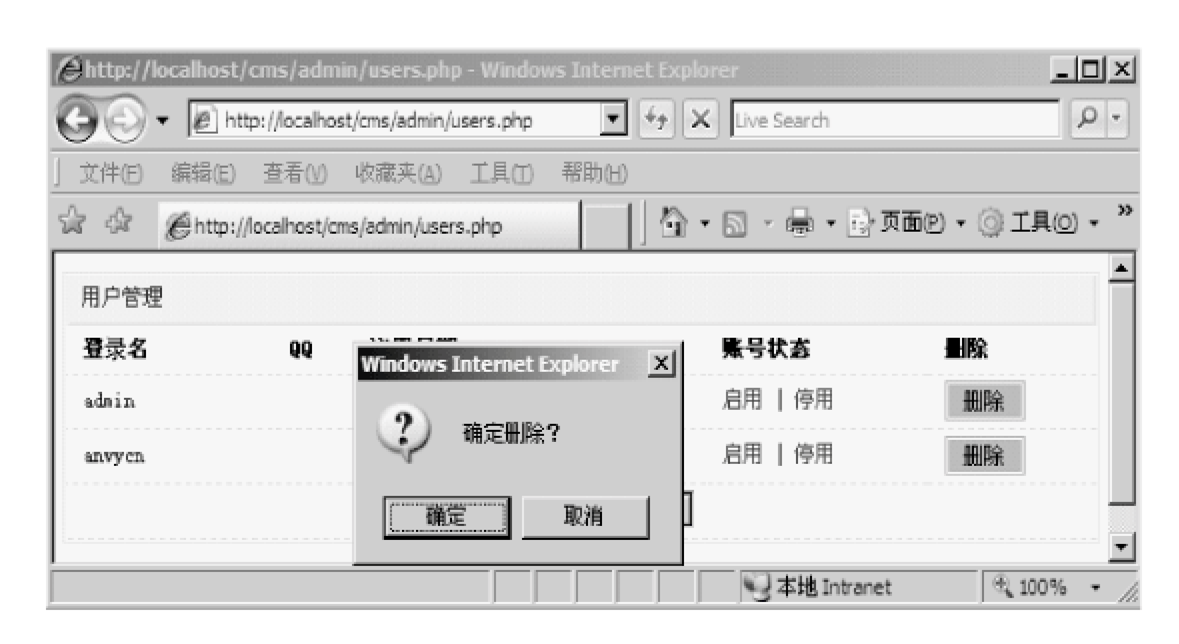


图 17.7　删除指定注册用户

17.5.2　文章管理

文章管理模块是内容管理系统的核心。传统的网站内容通常是以编辑后台发文章的形

式发布的，任何内容的修改都可以看作是对应的某个“文章”的修改，可以方便地通过文章编辑器修改。文章管理部分提供了对文章的基础管理操作，如新增文章、编辑修改、指定文章删除等。

1．文章列表

通过管理后台编辑，可以对已经录入的文章按照栏目分类进行浏览。在程序的实现上，是指定分类下的所有文章条目的遍历。

下面介绍文章管理列表页面的程序（articles.php）的代码实现：

```
<?php
include ('../init.inc.php');                          //数据库配置文件
$c = $_GET ['c'] ? $_GET ['c'] : $_POST ['c'];
//判断文章的分类
switch ($c) {
    case 1 :
        $cname = "黄金知识";
        break;
    case 2 :
        $cname = "资金评论";
        break;
    case 3 :
        $cname = "财经时事";
        break;
    case 4 :
        $cname = "研发报告";
        break;
    case 5 :
        $cname = "业内协会";
        break;
    case 6 :
        $cname = "会员文章";
        break;
}
usdb ( "articles" );                                  //初始化文章操作类
uslib ( "page" );                                     //初始化分页类
$page = $_GET ['page'] ? $_GET ['page'] : 1;
$page_size = 12;
//判断操作类型(删除)
if ($_GET ['del']) {
    $articles->del ( $_GET ['del'] );
    header ( "location:articles.php?page=$page&c=$c" );      //浏览器重定向
}
$strWhere = " AND Category = $c ORDER BY ID DESC";
$rows = $articles->getList ( $strWhere, $page_size, $page );//获得文章列表
$pager = new listPage ( $articles->recordNum, $page_size, 8 );
                                                      //初始化分页列表
$pager->lastPage = '<font face="webdings">7</font>';
$pager->nextPage = '<font face="webdings">8</font>';
$pager->firstPage = '<font face="webdings">9</font>';
$pager->endPage = '<font face="webdings">:</font>';
$pager->cssClass = '';
$pager->createPage ();
?>
```

对应的操作模板代码如下：

```
<table align=center cellspacing=0 cellpadding=0>
    <tr class=head>
        <td colspan="4"><?php echo $cname?></td>
    </tr>
    <tr class=line>
        <td><strong>日期</strong></td>
        <td><strong>标题</strong></td>
        <td><strong>修改</strong></td>
        <td><strong>删除</strong></td>
    </tr>
    <?php
foreach($rows as $row){
?>
    <tr class=line>
        <td width="120"><?php echo $row['Date']?></td>
        <td><?php echo $row['Title']?></td>
        <td width="60"><input name="Submit2" type="button" class="btn"
            value="修改" onclick="window.location= 'articles_save.php? ID=
            <?php
        echo $row['ID']?>&page=<?php echo $page?>&c=<?php echo $c?>'" />
        </td>
        <td width="60"><input name="Submit2" type="button" class="btn"
            value="删除" onclick="if(confirm( '确定删除? ')){window. location='?
            del=<?php
        echo $row['ID']?>&page=<?php echo $page?>&c=<?php echo $c?>';}"
        /></td>
    </tr>
    <?php
}
?>
    <tr class=line>
        <td colspan="4" align="center"><?php echo $pager->pageAll;?></td>
    </tr>
</table>
```

【代码解读】

在上面的代码通过判断传入的栏目 id，实现程序的流程控制。这里使用 switch()函数实现，代码模型如下：

```
//获得传入的栏目 id 变量
$c = $_GET['c'] ? $_GET['c'] : $_POST['c'];
//判断文章的分类
switch($c){
    case 栏目 id:
        $cname = "栏目分类名称";
    break;
    …//省略，读者可参照程序源代码
}
```

注意： 删除文章的操作也被包含在这个部分中，可以理解成为删除后重新生成文章列表的过程，获得动作 del 判断执行 articles 类中的删除（del）方法。

文章删除具体的实现代码如下：

```
//初始化文章操作类
usdb("articles");
```

```
//判断操作类型（删除）
if($_GET['del']){
    $articles->del($_GET['del']);
    header("location:articles.php?page=$page&c=$c");
}
```

通过上面两步后，设置基本的文章类的参数就可以调用 getList()方法获得文章列表的内容了，接下来生成分页部分的变量，代码如下：

```
uslib("page");                                //初始化分页类
$page = $_GET['page'] ? $_GET['page'] : 1;    //获得当前页码
$page_size = 12;                              //偏移量
$strWhere = " AND Category = $c ORDER BY ID DESC";
$rows = $articles->getList($strWhere,$page_size,$page);//获得文章列表
//分页列表相关变量设置
$pager = new listPage($articles->recordNum,$page_size,8);
$pager->lastPage = '<font face="webdings">7</font>';
$pager->nextPage = '<font face="webdings">8</font>';
$pager->firstPage = '<font face="webdings">9</font>';
$pager->endPage = '<font face="webdings">:</font>';
$pager->cssClass = '';
$pager->createPage();
```

在浏览器中执行下面的链接：

```
http://localhost/cms/admin/articles.php?c=1
```

CMS 后台文章管理列表的界面浏览效果，如图 17.8 所示。



图 17.8　后台文章管理列表

2. 编辑和新增文章

当需要调整已经录入的文章内容，编辑人员就需要使用文章管理中的“文章编辑”功能，重新调整已经录入的文章内容。编辑功能的程序实现思路是以文章的 id 号为条件，查找到数据库中 articles 表中对应的文章数据，然后整理成“文章编辑器”预设的格式。通过重新编辑文章内容后，再执行数据库的更新（update）操作，完成后台编辑指定文章的过程。下面具体介绍代码的实现。

```
<?php
include ('../init.inc.php');
$c = $_GET ['c'] ? $_GET ['c'] : $_POST ['c'];
//根据传入值设置 cname 变量
switch ($c) {
    case 1 :
        $cname = "黄金知识";
        break;
    case 2 :
        $cname = "资金评论";
        break;
    case 3 :
        $cname = "财经时事";
        break;
    case 4 :
        $cname = "研发报告";
        break;
    case 5 :
        $cname = "业内协会";
        break;
    case 6 :
        $cname = "会员文章";
        break;
}
usdb ( 'articles' );                              //创建数据库连接
uslib ( 'SinaEditor/sinaEditor.class' );          //加载编辑器类
if ($_POST) {
    if ($thisID = $_POST ['ID'])
        $articles->update ( $_POST );             //更新指定文章的内容
    else
        $thisID = $articles->add ( $_POST );      //新增文章操作(插入数据)
    header ( "location:articles.php?page=$_POST[page]&c=$c" );
}
$row = $articles->getVar ( $_GET ['ID'] );
?>
```

新增和重新编辑文章内容代码对应的模板代码，负责显示操作区域页面。其中，包括引入的“编辑器”部分核心代码如下：

```
<table align=center cellspacing=0 cellpadding=0>
    <form method="post"><input type="hidden" name="ID" value="<?php
    echo $row['ID']?>" /> <input type="hidden" name="Category" value="<?php
    echo $c?>" /> <input type="hidden" name="page" value="<?php
    echo $_GET['page']?>" />
    <tr class=head>
        <td colspan="2"><?php echo $cname?></td>
    </tr>
    <tr class=line>
```

```
        <td width="9%">标题</td>
        <td width="91%"><input name="Title" type="text" class="input"
            id="Title" size="70" value="<?php echo $row['Title']?>" /></td>
    </tr>
    <tr class=line>
        <td valign="top">内容</td>
        <td>
<?php
    $fck = new sinaEditor('Content');          //实例化编辑器
    $fck->BasePath  = '../libs/SinaEditor/' ;  //设置编辑器路径
    $fck->Value     = $row['Content'] ;        //获得编辑区内容
    $fck->Width = "100%";                      //设置显示宽度百分比
    $fck->Height = 500;                        //设置高度像素
    $fck->Create() ;
?>
        </td>
    </tr>
    <tr class=line>
        <td colspan="2" align="center"><input name="Submit" type="submit"
            class="btn" value="提交" /> <input name="Submit" type="button"
            class="btn" value="返回" onclick="history.go(-1);" /></td>
    </tr>
    </form>
</table>
```

【代码解读】

上面代码是以“文章编辑器”为核心的，先介绍如何调用“文章编辑器”，请看下面的调用代码：

```
<?php
    $fck = new sinaEditor('Content');          //实例化编辑器
    $fck->BasePath  = '../libs/SinaEditor/' ;  //设置编辑器路径
    $fck->Value     = $row['Content'] ;        //获得编辑区内容
    $fck->Width = "100%";                      //设置显示宽度百分比
    $fck->Height = 500;                        //设置高度像素
    $fck->Create() ;
?>
```

首先实例化编辑器设置类库的调用目录，然后设定编辑器的内容显示部分即：

```
$fck->Value     = $row['Content'] ;
```

再设定编辑器的显示尺寸信息即：

```
$fck->Width = "100%";
$fck->Height = 500;
```

【代码解读】

这里设定高度是 500 个像素，宽度为 100%。了解了编辑器的显示样式之后，程序就可以根据获得文章 id，从数据库中获取指定文章的详细数据，再赋值给$row 变量数组。这里需要注意的是，程序根据是否有表单传递文章来做流程控制，当$_POST['ID']有值时执行更新操作（update），否则认为是新增文章执行新增文章操作（add），代码如下：

```
$c = $_GET['c'] ? $_GET['c'] : $_POST['c'];
//根据传入值设置 cname 变量
switch($c){
```

```
    case 1:
        $cname = "黄金知识";
    break;
    …//频道栏目菜单
}
usdb ( 'articles' );                            //创建数据库连接
uslib ( 'SinaEditor/sinaEditor.class' );        //加载编辑器类
if($_POST){
    if($thisID = $_POST['ID'])
        $articles->update($_POST);              //更新指定文章的内容
    else
        $thisID=$articles->add($_POST);         //新增文章操作(插入数据)
    //新增后跳转
    header("location:articles.php?page=$_POST[page]&c=$c");
}
$row = $articles->getVar($_GET['ID']);
```

在文章列表页标题后单击“编辑”按钮，或者在浏览器中执行下面的链接：

```
http://localhost/cms/admin/articles_save.php?ID=7264&page=1&c=1
```

CMS 后台文章管理文章编辑界面浏览效果如图 17.9 所示。

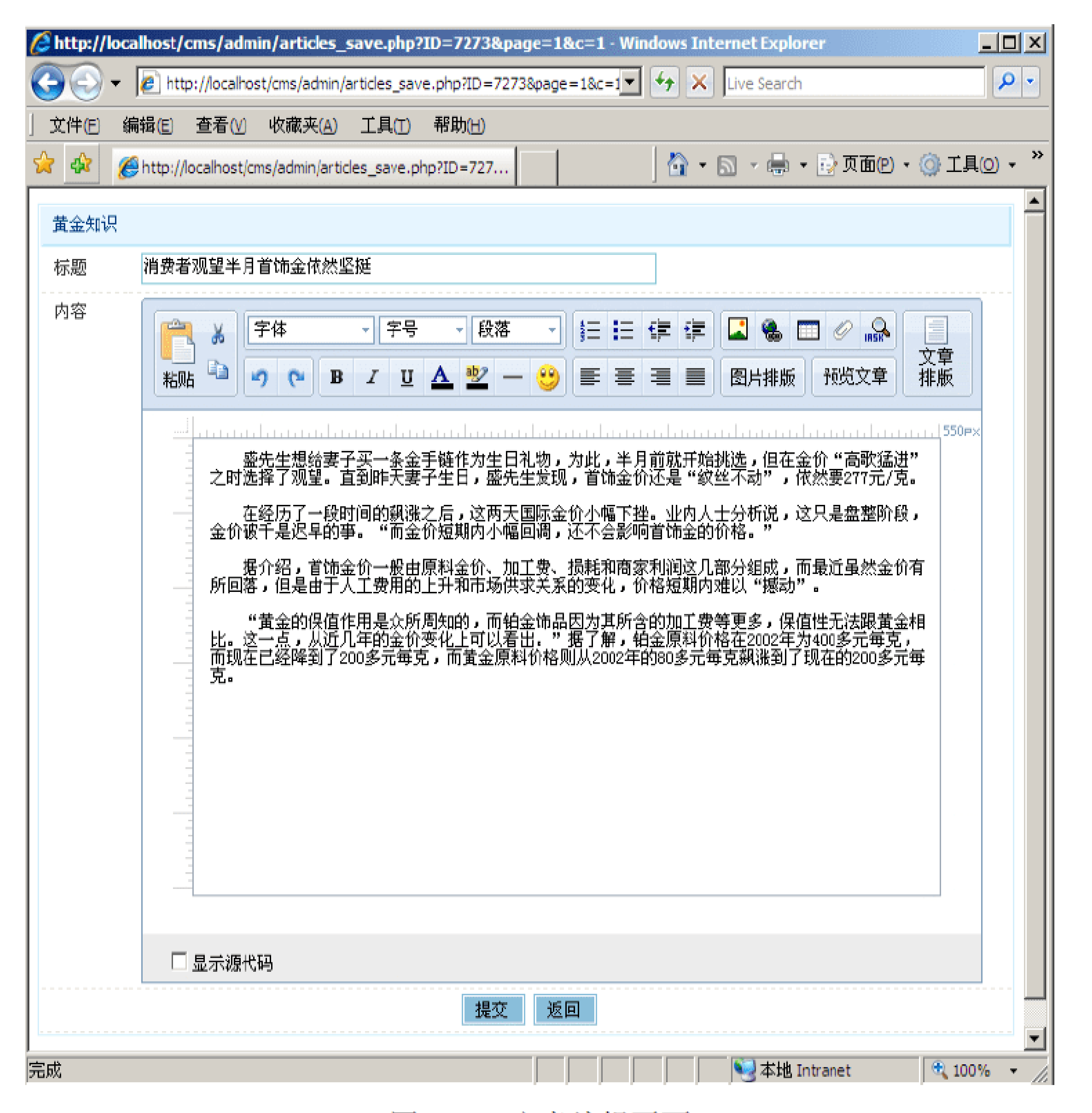


图 17.9　文章编辑页面

在文章管理栏目中单击“新增文章”按钮，或者在浏览器中执行下面的链接：

```
http://localhost/cms/admin/articles_save.php?c=1
```

CMS 后台文章管理新增文章界面，新添加的文章有两部分需要录入，即文章标题和内容代码。填入测试数据浏览效果如图 17.10 所示，单击“提交”按钮完成新文章录入。

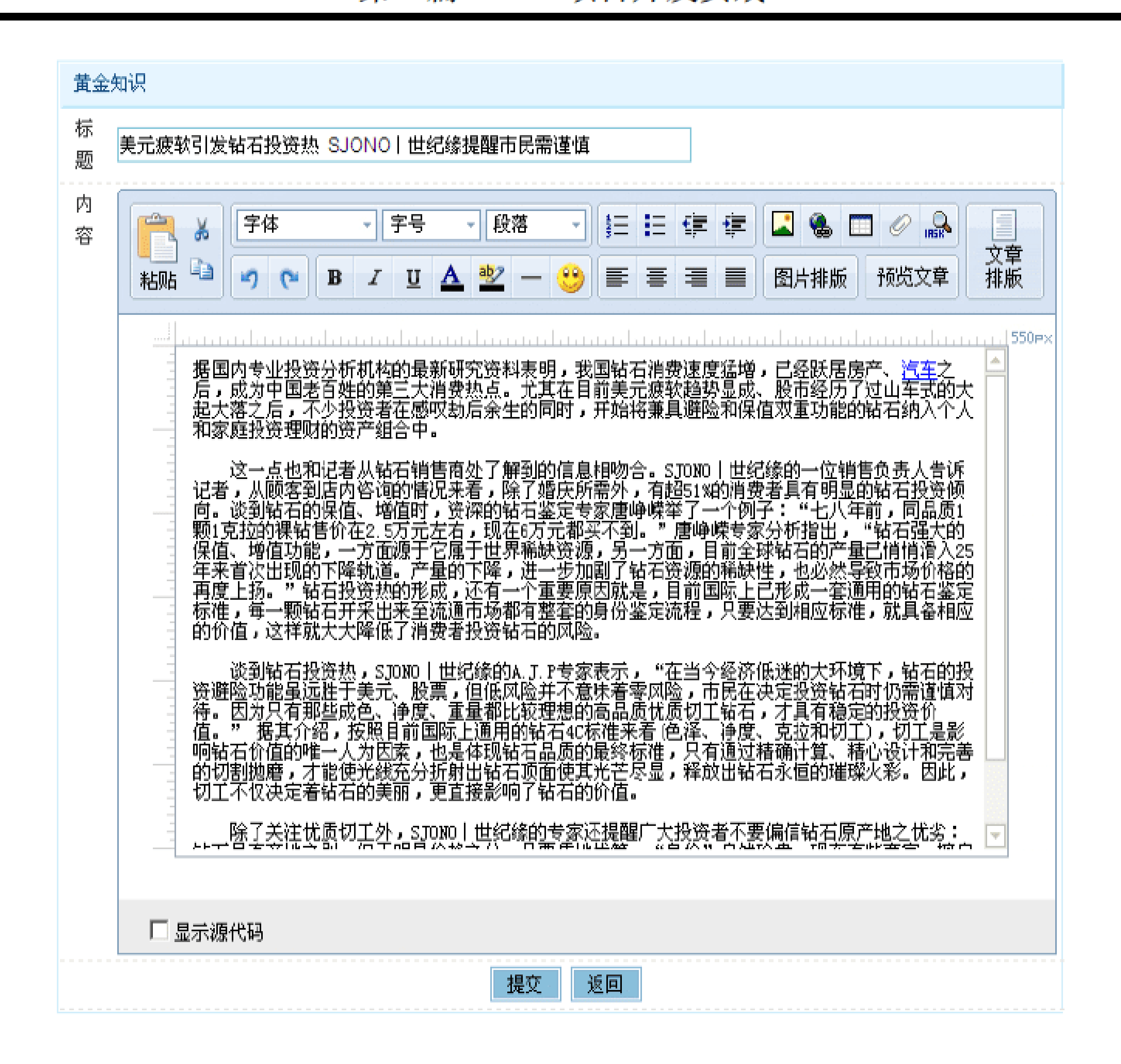


图 17.10　新增文章

17.5.3　网站相关栏目

网站相关栏目是管理网站相关信息的栏目，这些信息包括关于我们、联系方式、友情链接、修改管理员密码等操作。下面分别介绍程序是如何实现网站相关栏目及其子栏目的管理功能的。

1．关于我们

“关于我们”栏目是管理网站所有者信息的栏目，由于在前面的基础设计部分创建了处理网站相关信息的操作类，因此这部分的代码非常的简洁。读者在学习的时候，主要了解程序的流程控制和与表单页面交互部分的内容。“关于我们”的代码如下：

```
<?php
include ('../init.inc.php');
usdb ( 'site' );                        //创建数据库连接（site 表）
$Title = $_GET ['t'] ? $_GET ['t'] : $_POST ['Title'];
if ($_POST)
    $site->setVar ( $_POST );
    $row = $site->getVar ( $Title );    //获得项目名称
?>
    <form method="post"><input type="hidden" name="Title" value="<?php
    echo $Title?>" />
    <tr class=head>
        <td><?php echo $Title?></td>
    </tr>
    <tr class=line align="center">
        <td>
<?php
```

```
    uslib('SinaEditor/sinaEditor.class');
    $fck = new sinaEditor('Content');           //实例化编辑器
    $fck->BasePath  = '../libs/SinaEditor/' ;   //设置编辑器路径
    $fck->Value     = $row['Content'] ;         //获得编辑区内容
    $fck->Width = "100%";                       //设置显示宽度百分比
    $fck->Height = 500;                         //设置高度像素
    $fck->Create() ;
?>
<br />
        <input class="btn" value="确定提交" type="submit" /></td>
    </tr>
    </form>
```

【代码解读】

程序首先实例化 site 类，它里面封装了针对 site 表操作的具体方法，再根据传入的“动作”值（关于我们）作为查询条件从数据库中取出对应的内容。代码如下：

```
usdb('site');                                   //创建数据库连接（site 表）
$Title = $_GET['t'] ? $_GET['t'] : $_POST['Title'];
if($_POST)
    $site->setVar($_POST);
$row = $site->getVar($Title);                   //获得项目名称
```

在前面介绍过“文章编辑器”的调用样式，这里的使用方法完全一样。读者需要注意，调用时的路径设置和实例化的名称不要重复，调用编辑器的代码如下：

```
uslib('SinaEditor/sinaEditor.class');
$fck = new sinaEditor('Content');               //实例化编辑器
$fck->BasePath  = '../libs/SinaEditor/' ;       //设置编辑器路径
$fck->Value     = $row['Content'] ;             //获得编辑区内容
$fck->Width = "100%";                           //设置显示宽度百分比
$fck->Height = 500;                             //设置高度像素
$fck->Create() ;
```

在网站相关栏目中单击“关于我们”链接，或者在浏览器中执行下面的链接：

```
http://localhost/cms/admin/site.php?t=关于我们
```

效果如图 17.11 所示。

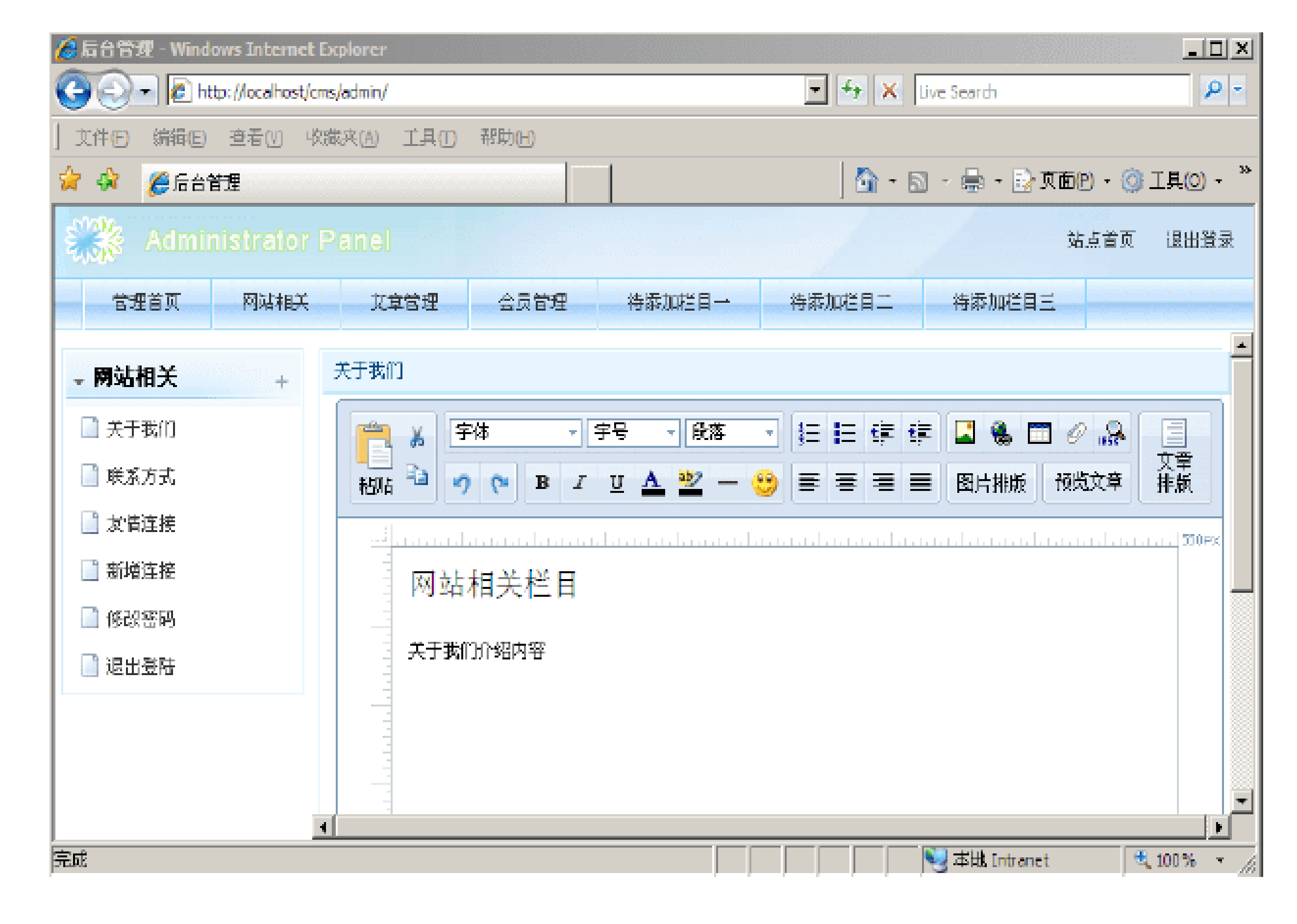


图 17.11　关于我们

2．联系方式

“联系方式”栏目是为了方便访问者与网站的管理员联系的栏目，这部分的管理是按照给定格式，编辑修改其中的内容，重点还是理解程序的控制流程和 HTML 表单的交互过程。“联系方式”栏目程序（service.php）的代码内容如下：

```
<?php
include('../init.inc.php');
usdb('service');                //创建数据库连接（service）
if($_POST)
    $service->update($_POST);   //更新操作
    $row = $service->get();
?>
<!--以下为页面模板部分-->
<table align=center cellspacing=0 cellpadding=0>
    <tr class=head>
        <td colspan="2">联系方式</td>
    </tr>
    <form name="" action="" method="post">
    <tr class=line>
        <td>地址</td>
        <td><input name="addr" type="text" class="input" value="<?php
        echo $row['addr']?>" size="70" /></td>
    </tr>
    <tr class=line>
        <td>邮箱</td>
        <td><input name="email" type="text" class="input" value="<?php
        echo $row['email']?>" size="60" /></td>
    </tr>
    <tr class=line>
        <td width="20%">电话</td>
        <td><input name="tel" type="text" class="input" value="<?php
        echo $row['tel']?>" size="60" /></td>
    </tr>
    <tr class=line>
        <td>传真</td>
        <td><input name="fax" type="text" class="input" value="<?php
        echo $row['fax']?>" size="60" /></td>
    </tr>
    <tr class=line>
        <td colspan="2" align="center"><input type="submit" value="确定修改"
            class="btn" /></td>
    </tr>
    </form>
</table>
</div>
```

【代码解读】

程序获得联系方式部分的内容信息后，再重新获得表单提交的信息。更新数据库中对应的字段内容，代码如下：

```
usdb('service');                //创建数据库连接（service）
if($_POST)
    $service->update($_POST);   //更新操作
    $row = $service->get();
```

在网站相关栏目中单击“联系方式”链接或者在浏览器中执行下面的链接：

```
http://localhost/cms/admin/service.php
```

浏览效果如图 17.12 所示。



图 17.12　联系方式

3．友情链接管理

“友情链接”管理功能是方便网站之前的合作，比如互换链接等操作。具体的管理功能为新增、查看、编辑、删除。

新增友情链接程序（links_save.php），通过合理的设计页面表单可以帮助管理员正确、快速输入友情链接的资料信息，该程序的代码内容如下：

```
<?php
include ('../init.inc.php');
usdb ( 'links' );                                   //创建数据库连接（links 表）
if ($_POST) {
    if ($thisID = $_POST ['ID'])                    //判断 id 是否存在
        $links->update ( $_POST );                  //更新操作
    else
        $thisID = $links->add ( $_POST );           //不存在则新增项目（友情链接）
    if ($_FILES ['file'] ['tmp_name']) {
        copy ( $_FILES ['file'] ['tmp_name'], '../uploads/' . ($fname =
        'links_' . $thisID . '.jpg') );
        $links->setPic ( $thisID, $fname );
    }
    header ( "location:links.php?page=$_POST[page]" ); //浏览器重定向
}
$row = $links->getVar ( $_GET ['ID'] );
?>
<!--友情链接管理模板及表单-->
<table align=center cellspacing=0 cellpadding=0>
    <form method="post" enctype="multipart/form-data">
    <input type="hidden"name="ID" value="<?php echo $row ['ID']?>" />
    <input type="hidden"name="page" value="<?php echo $_GET ['page']?>" />
    <tr class=head>
        <td colspan="2">友情连接</td>
    </tr>
```

```
    <tr class=line>
        <td width="9%">文字说明</td>
        <td width="91%">
        <input name="Title" type="text" class="input" id="Title" size="30"
        value="<?php echo $row ['Title']?>" />
        </td>
    </tr>
    <tr class=line>
        <td>连接 logo</td>
        <td><input name="file" type="file" class="input" />
  <?php
        echo $row ['Image'] ? "<img src=\"../uploads/$row[Image]\" />" : ''?>
        </td>
    </tr>
    <tr class=line>
        <td>连接地址</td>
        <td><input name="Url" type="text" class="input" id="Url" size="60"
            value="<?php
            echo $row ['Url']?>" /></td>
    </tr>
    <tr class=line>
        <td colspan="2"><input name="Submit" type="submit" class="btn"
            value="确认提交" /></td>
    </tr>
    </form>
</table>
```

【代码解读】

上面的代码判断是否有 id 传入控制程序的流程。当有 id 传入时，被认为是对已有的内容进行编辑维护操作（update）；没有 id 时，则被认为是新创建一条信息，执行新增（add）操作。代码如下：

```
<?php
include ('../init.inc.php');
usdb ( 'links' );                                //创建数据库连接（links 表）
if ($_POST) {
    if ($thisID = $_POST ['ID'])                 //判断 id 是否存在
        $links->update ( $_POST );               //更新操作
    else
        $thisID = $links->add ( $_POST );   //不存在则新增项目（友情链接）
    if ($_FILES ['file'] ['tmp_name']) {
        copy ( $_FILES ['file'] ['tmp_name'], '../uploads/' . ($fname =
        'links_' . $thisID . '.jpg') );
        $links->setPic ( $thisID, $fname );
    }
    header ( "location:links.php?page=$_POST[page]" ); //浏览器重定向
}
$row = $links->getVar ( $_GET ['ID'] );
?>
```

注意： 图片上传部分的处理使用 PHP 内置 copy()函数实现，读者在实际使用中要注意路径参数的写法问题，以免程序出错。

在网站相关栏目中单击“新增链接”链接，或者在浏览器中执行下面的链接：

```
http://localhost/cms/admin/link_save.php
```

浏览效果如图 17.13 所示。

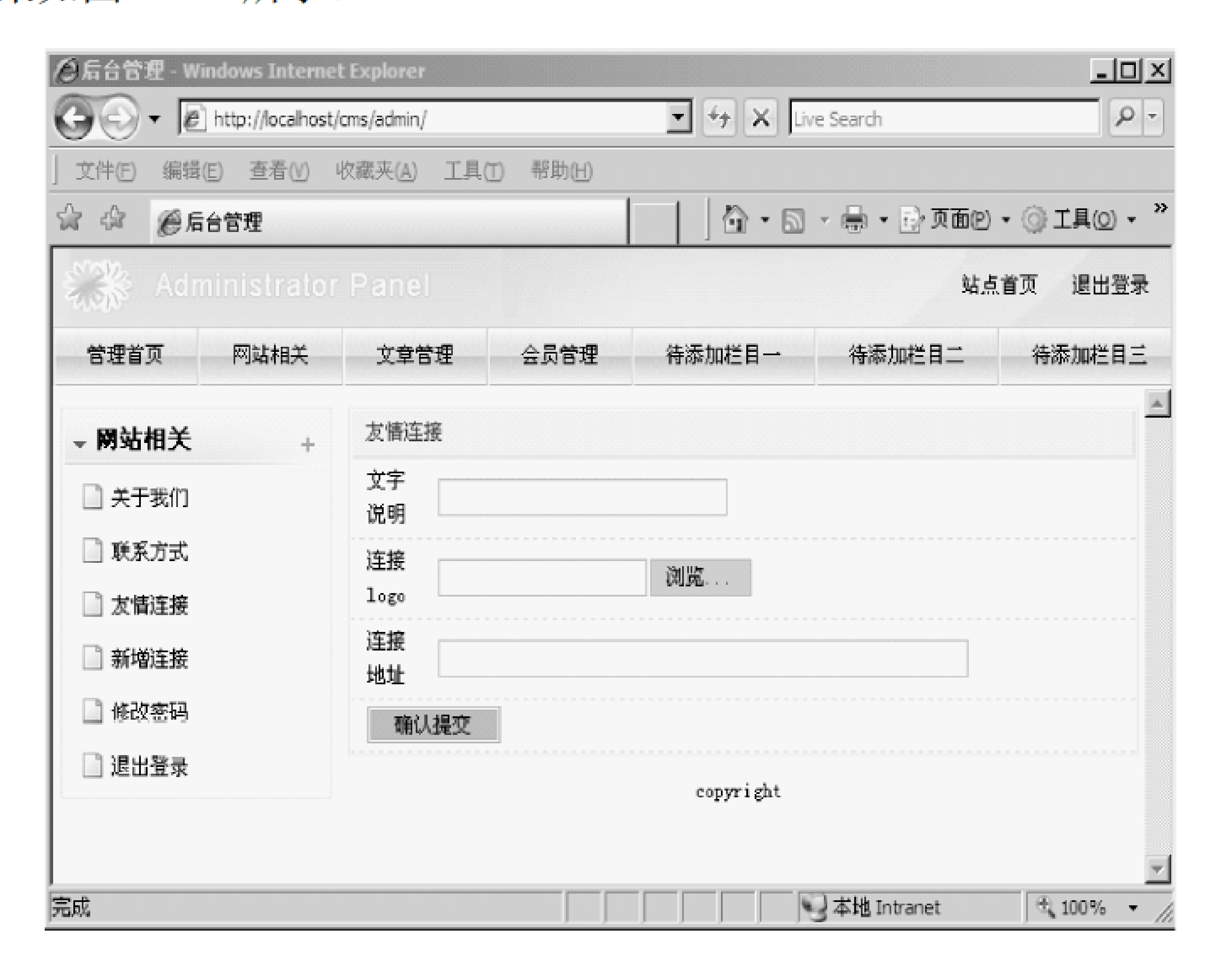


图 17.13　新增友情链接

查看已有的友情链接列表程序（link.php），代码如下：

```
    <?php
    include ('../init.inc.php');
    usdb ( "links" );                //创建数据库连接（links 表）
    uslib ( "page" );                //加载分页类
    $page = $_GET ['page'] ? $_GET ['page'] : 1;
    $page_size = 12;                 //设置偏移量长度
    //删除操作
    if ($_GET ['del']) {
        $links->del ( $_GET ['del'] ); //执行删除
        header ( "location:links.php?page=$page" );
    }
    //获得友情链接数据列表
    $rows = $links->getList ( $strWhere, $page_size, $page );
    $pager = new listPage ( $links->recordNum, $page_size, 8 );
    $pager->lastPage = '<font face="webdings">7</font>';
    $pager->nextPage = '<font face="webdings">8</font>';
    $pager->firstPage = '<font face="webdings">9</font>';
    $pager->endPage = '<font face="webdings">:</font>';
    $pager->cssClass = '';
    $pager->createPage ();
    ?>
<!--友情链接操作模板-->
<table align=center cellspacing=0 cellpadding=0>
    <tr class=head>
        <td colspan="5">友情链接</td>
    </tr>
    <tr class=line>
```

```
        <td><strong>文字</strong></td>
        <td><strong>Logo</strong></td>
        <td><strong>链接地址</strong></td>
        <td><strong>修改</strong></td>
        <td><strong>删除</strong></td>
    </tr>
    <?php
        foreach ( $rows as $row ) {//循环显示列表
    ?>
    <tr class=line>
        <td width="20%">
      <?php
        echo $row ['Title'];        //输出标题
    ?>
     </td>
        <td>
            <?php
                                    //输出链接图标
                echo $row ['Image'] ? "<img src=\"../uploads/$row[Image]\"
                />" : '文字链接'?>
        </td>
        <td>
        <?php
            echo $row ['Url'];      //输出链接地址
        ?>
        </td>
        <td><input name="Submit2" type="button" class="btn" value="修改"
            onclick="window.location='links_save.php?ID=<?php
        echo $row ['ID']?>&page=<?php
        echo $page?>'" /></td>
        <td><input name="Submit2" type="button" class="btn" value="删除"
            onclick="if(confirm('确定删除？')){window.location='?del=<?php
        echo $row ['ID']?>&page=<?php
        echo $page?>';}" /></td>
    </tr>
    <?php
    }
    ?>
    <tr class=line>
        <td colspan="5" align="center">
        <?php
            echo $pager->pageAll;   //输出分页相关内容
        ?>
        </td>
    </tr>
</table>
</div>
```

【代码解读】

上面的代码使用 link 类中封装的方法，代码简洁。其中，获得“友情链接”列表的语句如下：

```
$rows = $links->getList($strWhere,$page_size,$page);
```

然后通过 Page 类设置了分页变量和参数，具体设置如下：

```
$pager = new listPage($links->recordNum,$page_size,8);
$pager->lastPage = '<font face="webdings">7</font>';  //上一页
$pager->nextPage = '<font face="webdings">8</font>';  //下一页
$pager->firstPage = '<font face="webdings">9</font>'; //第 1 页
$pager->endPage = '<font face="webdings">:</font>';   //最后一页
$pager->cssClass = '';
$pager->createPage();
```

在网站相关栏目中单击“友情链接”链接，或者在浏览器中执行下面的链接：

```
http://localhost/cms/admin/links.php
```

浏览效果如图 17.14 所示。



图 17.14　友情链接列表

当需要对某条“友情链接”的内容做修改时，选择对应的“修改”按钮，程序会跳转到 links_save.php 程序处理修改信息并更新数据库中相应的字段。“友情链接”编辑界面的如图 17.15 所示。



图 17.15　友情链接内容编辑

4．修改管理员密码

网站的管理员拥有最高的管理权限，因此没有把管理员账户放在会员管理中保存，而是单独作为网站相关栏目中的一个单独的项目。管理员密码修改程序（pwd.php）的具体代码如下：

```
<?php
include ('../init.inc.php');
usdb ( 'master' );                          //创建数据库连接（master 表）
if ($_POST) {
    if ($_POST ['Password'] != 'demo') //判断密码是否存在
        $_POST ['Password'] = md5 ( $_POST ['Password'] ); //密码校验
    else
        unset ( $_POST ['Password'] );                   //注销密码变量
    $master->update ( $_POST );
}
$row = $master->getVar ( $_SESSION ['admin_id'] );
?>
```

管理员密码修改程序对应模板部分的代码如下：

```
<table align=center cellspacing=0 cellpadding=0>
    <form method="post">
    <input type="hidden" name="ID"value="<?php echo $row ['ID']?>" />
    <tr class=head>
        <td colspan="2">修改密码</td>
    </tr>
    <tr class=line>
        <td width="9%">登入账号</td>
        <td width="91%">
        <input name="UserName" type="text" class="input" id="Title" size=
        "30" value="<?php echo $row ['UserName']?>" /></td>
    </tr>
    <tr class=line>
        <td>登入密码</td>
        <td><input name="Password" type="password" class="input" value="demo" />
        </td>
    </tr>
    <tr class=line>
        <td colspan="2"><input name="Submit" type="submit" class="btn"
            value="确认提交" /></td>
    </tr>
    </form>
</table>
```

【代码解读】

以上代码使用了 master 类中的两个方法来处理管理员账户修改的操作，即 update()方法用来更新账户密码，getVar()方法用来获得当前管理员的信息。

```
$master->update($_POST);
$row = $master->getVar($_SESSION['admin_id']);
```

在网站相关栏目中单击“修改密码”链接，或者在浏览器中执行下面的链接：

```
http://localhost/cms/admin/pwd.php
```

浏览效果如图 17.16 所示。

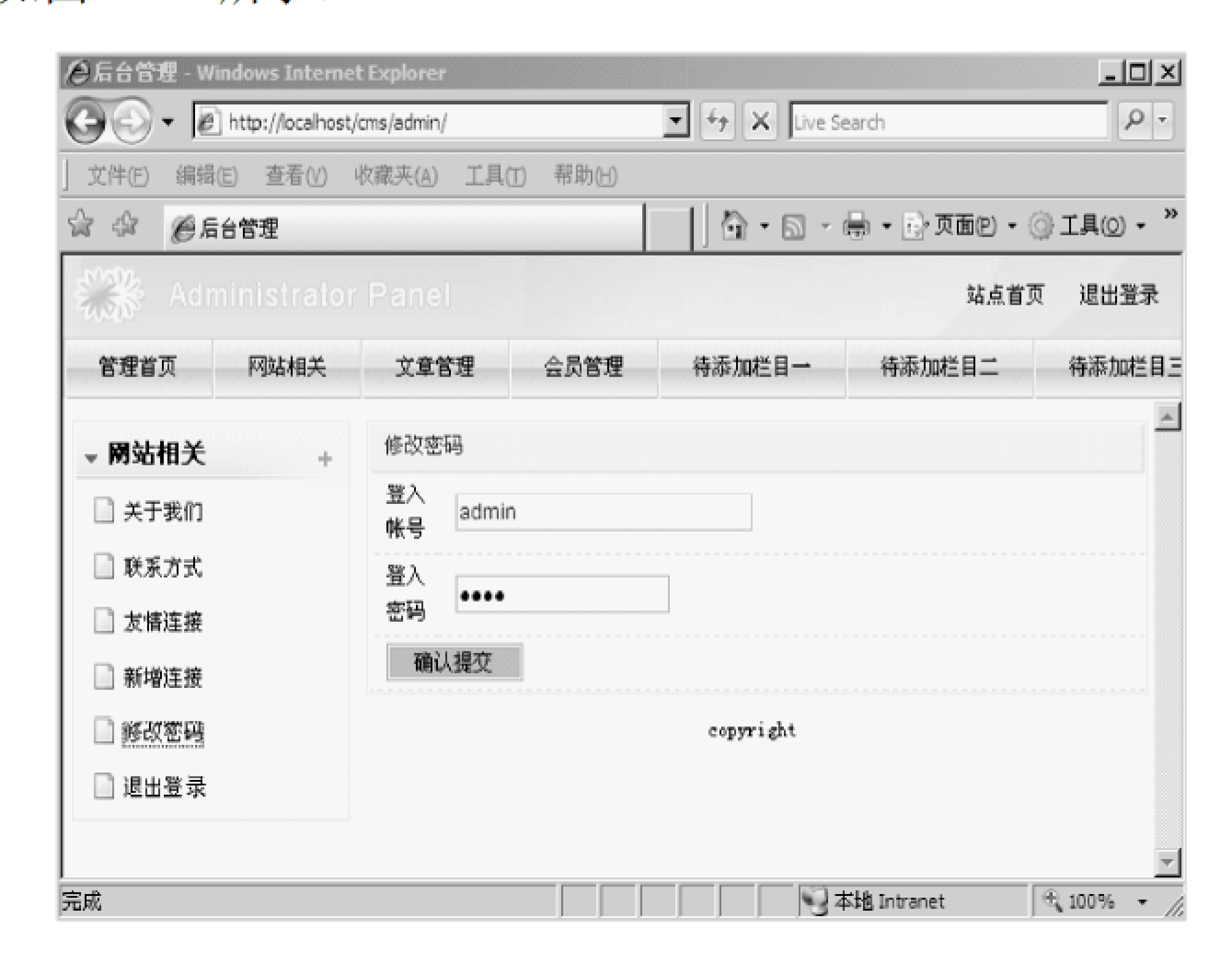


图 17.16　管理员账户修改

17.6　小　　结

内容管理系统（CMS）模块是使用 PHP 搭建后台系统的一个典型应用。读者需要了解在 PHP 中使用面向对象设计（Object Oriented Programming，OOP）的开发方式编程。

本章从 CMS 设计基础讲起，先介绍了 CMS 基本概念，然后详细说明了 CMS 主要实现功能。在明确了实现功能之后，介绍了内容管理系统数据库的设计，详细讲解了涉及的表和表的功能。在代码实现的部分从如何管理框架搭建开始，首先引导读者搭建一个管理的框架，然后创建了基础类库，再通过这几个基础的类库不断地添加功能模块，即可将各个后台管理功能逐一实现。

第 18 章　娱乐联播（RSS 内容聚合+Magpie 抓取）

娱乐联播模块是一个典型的 RSS 内容聚合的应用。本章主要介绍了 PHP 在 RSS 内容聚合方面的基础原理和实现的具体方法。通过本章的学习，读者可以了解到一个完整的 RSS 应用的构成。在技术层面读者可以了解到，如何通过 PHP 抓取类的形式解析、获取信息源中的内容信息。

本章主要涉及的知识点如下所示。

- 递归：调用自身的编程技巧称为递归。
- RSS 数据源：学会查询、获取数组中的单一数据，让数据为我所用。
- RSS 内容聚合：学会逐个浏览数组中的单一数据。
- mb_internal_encoding()：设计需要截取字符串的编码，括号中的参数为具体的编码名称。

18.1　娱乐联播基础设计

娱乐联播是一个基于 RSS 新闻聚合器的典型应用。在基于 RSS 的众多应用中，新闻聚合是最为常见的，也是最具代表性的。读者可以通过本节，了解 RSS 的基本概念和搭建 RSS 应用程序的的基本方法和步骤。

18.1.1　RSS 基本概念

RSS 也叫聚合 RSS，是在线共享内容的一种简易方式（也叫聚合内容，Really Simple Syndication）。通常在获取时效性较强内容的应用上使用 RSS 订阅能更快速获取信息，网站提供 RSS 输出，有利于让用户获取网站内容的最新更新。在开始设计之前先介绍几个 RSS 的基本概念。

（1）RSS 数据源：目标网站发布一个 RSS 文件（一般称为 RSS Feed）后，这个 RSS Feed 中包含的信息就能直接被其他站点调用。由于这些数据都是标准的 XML 格式，所以也能在其他的终端和服务中使用，如 PDA、手机、邮件列表等。

（2）RSS 内容聚合：本章中介绍的 RSS 内容聚合是指通过定制不同的主题的新闻源的内容，将这些内容划分到对应的网站栏目中显示。读者可以通过一个页面了解到来自不同新闻源的某一个主题的内容。

RSS 是通过 XML 格式文件承载数据源实现数据交换。以下是 RSS2.0 格式的代码样式，

其中具体介绍 XML 格式新闻源中标签的含义。

```
<?xml version="1.0"?>
<!-- RSS 版本 -->
<rss version="2.0">
<!-- 以下为频道信息及新闻列表 -->
<channel>
<!-- 频道总体信息：开始 -->
<!-- 频道标题 -->
<title>Lift Off News</title>
<!-- 频道链接的总地址 -->
<link>http://liftoff.msfc.nasa.gov/</link>
<!-- 频道描述文字 -->
<description>Liftoff to Space Exploration.</description>
<!-- 频道使用的语言（zh-cn 表示简体中文） -->
<language>en-us</language>
<!-- 频道发布的时间 -->
<pubDate>Tue, 10 Jun 2003 04:00:00 GMT</pubDate>
<!-- 频道最后更新的时间-->
<lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>
<docs>http://blogs.law.harvard.edu/tech/rss</docs>
<!-- 频道生成器 -->
<generator>Weblog Editor 2.0</generator>
<managingEditor>editor@example.com</managingEditor>
<webMaster>webmaster@example.com</webMaster>
<ttl>5</ttl>
<!-- 频道总体信息：结束 -->
<!-- 每条 RSS 新闻信息都包含在 item 节点中-->
<item>
<!-- 新闻标题 -->
    <title>Star City</title>
<!-- 新闻链接地址 -->
    <link>http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp</link
>
<!-- 新闻内容简要描述 -->
    <description>How do Americans get ready to work with Russians aboard the
    International Space Station? They take a crash course in culture, language
    and protocol at Russia's Star City.</description>
<!--新闻发布时间 -->
    <pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>
<!--新闻目录 -->
    <category>IT</category>
<!--新闻作者 -->
    <author>bill</author>
    <guid>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>
</item>
</channel>
</rss>
```

18.1.2　娱乐联播流程

基于 RSS 的娱乐联播模块程序基本流程是首先获得娱乐联播内容，需要选择一种 RSS 阅读方式，可以通过给出外部链接跳转的方式也可以安装 RSS 的阅读器。然后将提供 RSS 服务的网站的数据接口链接，加入对应的频道即可。

具体步骤如下：

（1）在系统中录入的指定 RSS 信息源，并制定采集规则。

（2）启动服务端 RSS 采集程序，定时采集指定站点的 RSS 内容，按照设定的规则写入数据库。

（3）编写前台展示程序，按照对应的栏目从数据库中提取采集内容。

下面通过流程图为读者直观地介绍程序的执行流程。娱乐联播程序的执行流程如图 18.1 所示。

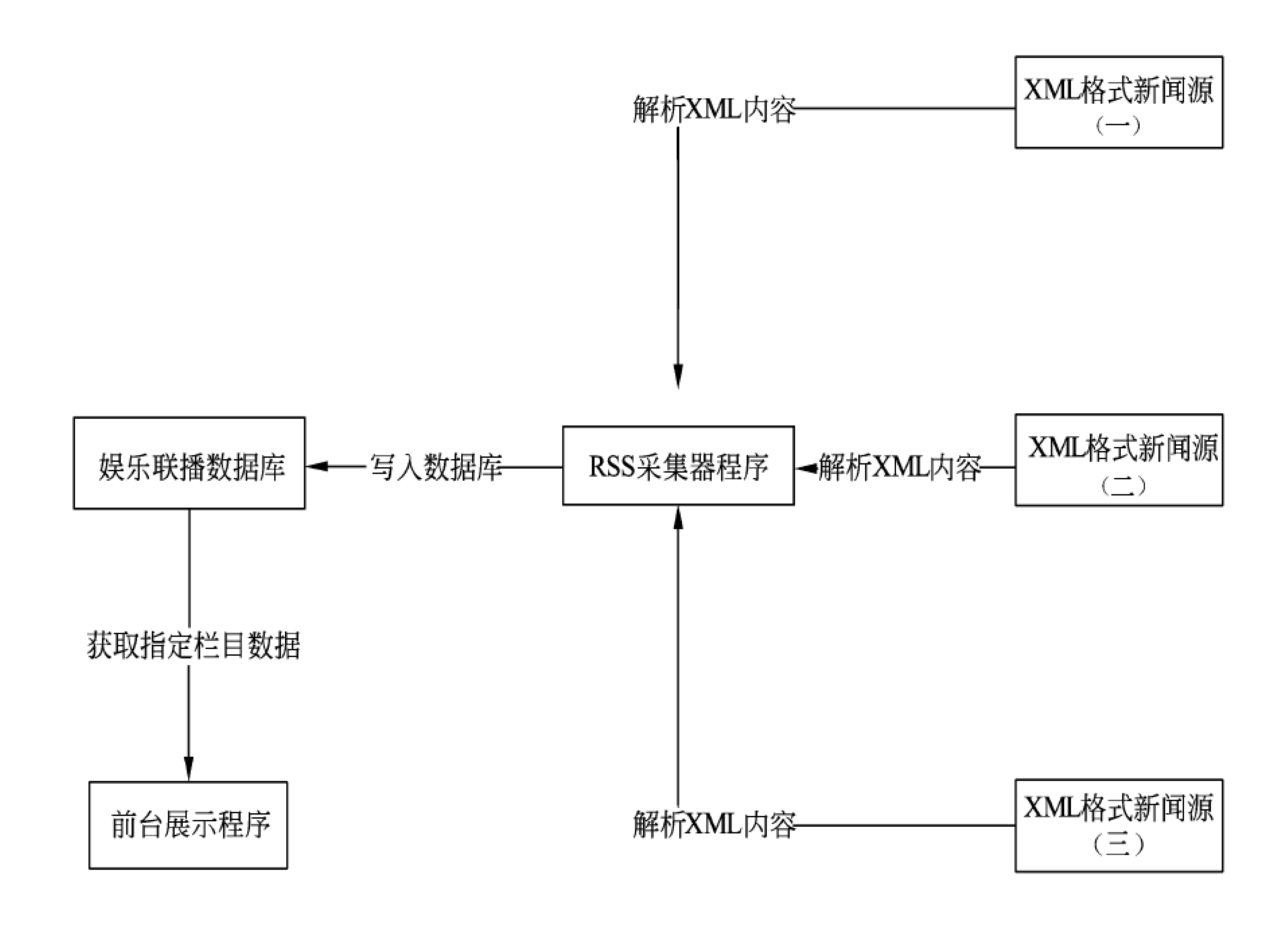


图 18.1　娱乐联播程序流程图

18.2　数据库设计

娱乐联播模块的设计目的是存储文章的分类、保存采集规则和数据源、保存采集新闻的内容。娱乐联播模块的数据库中用到了 3 张表，分别是栏目分类表（rss_catalog）表、RSS 数据源（rss_source）表、采集新闻内容（rss_content）表。

CMS 内容管理系统数据库表含义如下。

- 栏目分类（rss_catalog）表：存储栏目分类信息（栏目名称等）。
- RSS 数据源（rss_source）表：存储 RSS 数据源的详细信息。
- 采集新闻内容（rss_content）表：存储采集的新闻数据的详细信息。

通过上面介绍的 CMS 内容管理系统数据库表的功能划分，已经为每个表规定了表的含义和实现功能，下面介绍具体的设计步骤。

创建名为 lb_rss 的数据库，SQL 语句如下：

```
CREATE DATABASE 'lb_rss';
```

18.2.1　栏目分类表

在 lb_rss 数据库中，建立一个命名为 rss_catalog 的栏目分类表，结构如表 18.1 所示。

表 18.1　栏目分类表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
id	int	11	Not null	auto_increment	栏目编号（自增主键）
cataname	varchar	255	Not null		栏目名称
parentid	int	11	Not null	0	上级栏目 id
keywordname	varchar	255		Null	栏目关键字
urlname	varchar	255		Null	链接名称
cata_status	tinyint	1	Not null	0	栏目状态

栏目分类表是一个基础的存储栏目分类相关信息的表。它本身可以作为一个独立的表，也可以作为文章内容表的附表，在扩展方式上可以通过增加附表的方式实现。

为了便于后面程序的调试，向数据库中插入两条测试数据。插入测试代码的 SQL 语句如下：

```
INSERT INTO 'rss_catalog' VALUES (1, '娱乐联播', 0, '', '', 1);
INSERT INTO 'rss_catalog' VALUES (2, '明星', 1, '', '', 1);
INSERT INTO 'rss_catalog' VALUES (3, '命理', 1, '', '', 1);
```

18.2.2　RSS 数据源表

在 lb_rss 数据库中，建立一个命名为 rss_source 的 RSS 数据源表，结构如表 18.2 所示。

表 18.2　RSS数据源表

字　段	数据类型	长度	NULL	默 认 值	字 段 说 明
id	int	11	Not null	auto_increment	数据源编号（自增主键）
catalogid	varchar	11	Not null		数据源所属分类
rss_url	varchar	255	Not null		采集数据源地址
source	varchar	128	Not null		数据源名称
author	varchar	128	Not null		作者
update_time	datetime		Not null	0000-00-00 00:00:00	数据更新时间
get_time	int	8	Not null	0	采集时间
rss_status	tinyint	1	Not null	0	数据源状态

RSS 数据源表的设计目的是为了存储 RSS 数据源和设置采集参数，内容采集程序直接通过该表获得需要采集的数据源。

为了便于后面程序的调试，向数据库中初始化数据源数据。插入测试代码的 SQL 语句如下：

```
LOCK TABLES 'rss source' WRITE;
INSERT INTO 'rss_source' VALUES (13,'17','http://rss.sina.com.cn/news/allnews/
```

```
ent.xml','新浪','wzq','2007-03-30
16:06:57',100,0),(7,'4','http://rss.business.sohu.com/rss/waihuijiaoyi.
xml','','wzq','2008-10-17
15:30:05',100,1),(6,'2','http://rss.business.sohu.com/rss/yinhanglicai.
xml','','wzq','2008-10-17
15:30:05',100,1),(5,'3','http://rss.business.sohu.com/rss/licaishenghuo
.xml','','wzq','2008-10-17
15:30:05',100,1),(8,'12','http://rss.business.sohu.com/rss/baoxianlicai
.xml','','wzq','2008-10-17
15:30:05',100,1),(9,'13','http://rss.business.sohu.com/rss/gushibobao.x
ml','','wzq','2008-10-17
15:30:05',100,1),(10,'14','http://rss.business.sohu.com/rss/qihuoyaowen
.xml','','wzq','2008-10-17
15:30:04',100,1),(11,'15','http://news.baidu.com/n?cmd=1&class=autonews
&tn=rss&sub=0','','wzq','2008-10-17
15:30:04',100,1),(12,'16','http://rss.cnfol.com/1537.xml','','wzq','2008-10-17
15:30:04',100,1),(14,'15','http://news.baidu.com/n?cmd=1&class=housenew
s&tn=rss&sub=0','','wzq','2008-10-17
15:30:04',100,1),(15,'13','http://rss.business.sohu.com/rss/jijinshuju.
xml','','wzq','2008-10-17
15:30:04',100,1),(16,'3','http://rss.cnfol.com/1553.xml','','wzq','2008-10-17
15:30:04',10,1),(17,'3','http://rss.cnfol.com/1554.xml','','wzq','2008-
10-17 15:30:04',10,1);
UNLOCK TABLES;
```

18.2.3　采集新闻内容表

在 lb_rss 数据库中，建立一个命名为 rss_content 的采集新闻内容，结构如表 18.3 所示。

表 18.3　采集新闻内容表

字　段	数据类型	长度	NULL	默　认　值	字 段 说 明
id	int	11	Not null	auto_increment	文章编号（自增主键）
catalogid	varchar	11	Not null	0	文章所属分类
title	varchar	255	Not null		文章标题
url	varchar	255	Not null		该文章的链接
description	text	128	Not null		文章描述
source_id	int	8	Not null	0	数据源 id
source	varchar	255	Not null		数据来源
content_time	datetime		Not null	0000-00-00 00:00:00	采集时间
content_nums	int	12	Not null	0	文章内容条数
index_status	int	1	Not null	0	索引状态
content_status	tinyint	1	Not null	0	文章状态

采集新闻内容表的设计目的是为了存储采集程序从 RSS 数据源中采集的文章内容信息。

注意：如果该文章有更新，则程序会重复抓取并覆盖该条记录。

18.3　核心函数说明

本章中由于涉及的业务逻辑都是相似的，因此将复用程度很高的部分独立出来。通过

核心函数库的方式提供给其他程序调用，便于后续业务逻辑快速开发。创建名为 include.rssfuns.php 的函数库文件，用来存储这些函数，后续的程序也会以此实现具体的业务流程。

18.3.1　使用 get_ids()函数获得指定分类下的所有子分类 ID

该函数的功能是获得指定分类下的所有子分类 ID，以字符串的形式返回。函数代码如下：

```
<?php
//取得指定分类的所有子分类 ID 的函数
function get ids($id, $mysql)
{
    $sql = "select id from rss catalog where parentid =" . $id;
    $result = mysql_query ( $sql, $mysql );
    $return sets = $id;                              //递归变量 id
    while ( $tmp sets = mysql fetch array ( $result ) ) {
        $temp = get ids ( $tmp sets ["id"], $mysql );//获得当前 id 下的子分类 id
        if ($temp == "")
            $return set .= $temp;
        else
            $return_sets .= "," . $temp;
    }
    return $return_sets;                             //返回子分类 id 串
}
?>
```

【代码解读】

具体的操作是对数据库中 rss_catalog 表中存储的栏目分类信息进行变量，然后对获取的 id 进行整理。

18.3.2　使用 index_list()函数获取指定分类下的 RSS 抓取内容

该函数的功能是获取指定分类下的 RSS 抓取内容，并以数组列表的形式返回。函数代码如下：

```
<?php
//RSS 内容列表
function index_list($catalogid, $offset, $nums, $mysql) {
    $sql = "select id,title,url,source id,source ,content time from rss
    content " . "where  content_status=1 and catalogid in (" . $catalogid . ")
    " . "and index status='1' ORDER BY id desc LIMIT " . $offset . "," .$nums;
    $result = mysql query ( $sql, $mysql );
    $nums = mysql num rows ( $result );     //获得结果集的行数
    $articleList = "";
    for($i = 0; $i < $nums; $i ++)          //循环获得内容生成列表数组
    {
        $articleList [$i] = mysql_fetch_array ( $result );
    }
    return $articleList;                    //返回数组数据
```

```
}
?>
```

【代码解读】

具体的操作是对数据库中 rss_content 表中存储的已抓取的内容信息进行匹配，然后通过 for 循环函数将内容取出，整理成数组。

18.3.3　使用 content_info()函数获取指定 ID 下的信息

该函数的功能是获取指定 ID 下的信息内容，并以数组列表的形式返回。该函数代码如下：

```
<?php
//取出指定 ID 的抓取内容信息
function content_info($id, $mysql) {
    $sql = "select * from rss_content where id='" . $id . "' and content_
    status='1'";
    $result = mysql_query ( $sql, $mysql );
    $content [0] = mysql_fetch_array ( $result );  //获取指定内容
    return $content [0];                           //返回内容信息
}
?>
```

【代码解读】

具体的操作是对数据库 rss_content 表中指定 id 的信息匹配条件，然后从数据库中取出指定内容，整理成数组返回。

18.4　娱乐联播后台管理

本节将会介绍娱乐联播后台管理程序需要实现的管理功能，以及具体的代码实现。娱乐联播管理后台最主要的目的是，实现对 RSS 新闻源的基本设置功能和对已抓取新闻的一个基本的管理。具体的功能划分如下。

- ❑ RSS 栏目管理：设置娱乐联播下属的子栏目，即下属的子频道命名及从属关系。
- ❑ 添加 RSS 源：设置 RSS 来源的基本参数包括所属栏目、RSS 来源、RSS 地址、抓取间隔时间、RSS 抓取状态、编辑。
- ❑ 已有 RSS 源管理：管理已有的 RSS 源管理功能包括编辑（重置所有 RSS 参数）、删除功能。
- ❑ RSS 头条管理：推荐至首页头条 RSS 抓取信息可以设置排序和删除。
- ❑ RSS 信息源抓取：启动 RSS 抓取引擎从预先设置的新闻源抓取 RSS 新闻信息，并整理写入数据库。

18.4.1　RSS 栏目管理

RSS 栏目管理的作用是通过创建和管理 RSS 的栏目（子频道），便于管理员完成栏目

（频道）的规划和设计，这部分的重点是理解栏目的“树”的实现。RSS 栏目管理按照功能又分成栏目显示、子栏目添加、栏目修改和删除 4 个部分。下面逐一介绍这几个功能是如何实现的。

1．栏目树显示

栏目数显示部分是通过遍历数据中的栏目字段，根据 id 的从属关系将栏目以树形结构的方式显示出来。创建一个用来显示栏目树的 PHP 脚本文件，命名为 rss_catalog.php。该文件代码如下：

```
<?php
    //详细信息：显示分级树型结构，显示增加删除修改栏目
    include_once( dirname(__FILE__)."/include.smarty.php");
    include_once( dirname(__FILE__)."/include.mysqlconfig.php");
    include_once( dirname(__FILE__)."/include.rssfuns.php");
    //登录状态验证（可以根据实际情况使用 Session 变量实现）
    $login='already';
    if (!empty($login))
    {
        //smarty 标签变量，显示所有的 table
        $list = "";
        //根目录的父 id
        $id = 0;
        $mysql=mysql_connect($mysqlconf["host"],$mysqlconf["user"],$mysqlconf
        ["pass"]);
        mysql_select_db($mysqlconf["cmsdb"]);
        //获得栏目树
        $list = rss_catalogTree($id,$mysql);
        mysql_close();
        $smarty -> assign("list",$list);              //显示所有的条目
        $smarty -> display("./rss_catalog.tpl");
    }
    else
    {
        echo "<script language=\"javascript\" type=\"text/javascript\">
        alert(\"您没有权限进行此操作，请与管理员联系！\");history.back();</script>";
        header("Location: index.php?error_msg=您没有权限操作" );//浏览器重定向
        exit();
    }
?>
```

【代码解读】

栏目树显示部分由公共函数库中定义的 rss_catalogTree()函数实现。该函数的实现原理是通过递归的方式遍历父 id 下面的所有栏目，然后通过 for()循环生成带栏目标题的 HMTL 标签，再通过 rss_isCatalog()函数判断本栏目下是否有子栏目。请看下面两个函数的代码实现：

```
//判断本栏目下是否有子栏目
function rss_isCatalog($id,$mysql)
{
    $sql = "select count(*) as cnum from rss_catalog where parentid=" . $id;
```

```
    $result=mysql_query($sql,$mysql);
    $num = mysql_fetch_array($result);
    if($num[0] == 0)          //没有则返回假
        return false;
    else
        return true;          //有则返回真
}
//遍历显示树状目录，根据父 id 显示其下所有内容
function rss_catalogTree($id,$mysql)
{
    $list="";
    $list.="<table border=\"0\" cellspacing=\"0\" cellpadding=\"0\">";
    $sql = "select * from rss_catalog where parentid=" . $id . " order by id";
                                                     //查出所有父 id 为根目录的选项
    $result=mysql_query($sql,$mysql);
    $num=mysql_num_rows($result);                    //获得记录集行数
    for($i=0;$i<$num;$i++)
    {
        $arr[$i]=mysql_fetch_array($result);               //获得记录数组
        if(rss_isCatalog($arr[$i]["id"],$mysql))           //如果本栏目是目录
        {
            $list .="<tr><td id=\"catam". $arr[$i]["id"] ."\" class=\"emenu\"
            onMouseUp=\"colorChange(". $arr[$i]["id"].");document. getEl-
            ementById('lab').innerText='".$arr[$i]["cataname"]."'; hrefChan
            (". $arr[$i]["id"].");with(findObj('". "cata" . $arr[$i]["id"] ."'))
            if(style.display=='none'){style.display='';this.style. back-
            grounddImage='url(images/open.gif)';}else{style. display='none';
            this.style.backgroundImage='url(images/close.gif)';}\">". $arr
            [$i]["cataname"] ."</td></tr>";
        if($arr[$i]["id"] == 1 )
                $list .= "<tr id=\"". "cata" . $arr[$i]["id"] ."\">";
            Else
            //将标签赋值给 list 变量生成变量数组
            $list .= "<tr id=\"". "cata" . $arr[$i]["id"] ."\" style=\"display:
            none;\">";
            $list .= "<td class=\"submenu\">";
            $list .= rss_catalogTree($arr[$i]["id"],$mysql);
            $list .= "</td>";
            $list .= "</tr>";
        }
        else                                    //如果不是目录的显示方式
        {
            $list .="<tr><td id=\"catam". $arr[$i]["id"] ."\" class=\"file\"
            onMouseUp=\"colorChange(". $arr[$i]["id"] .");document. Get Elem-
            entById('lab').innerText='". $arr[$i]["cataname"] ."'; hrefChan
            (". $arr[$i]["id"] .");\">". $arr[$i] ["cataname"] . "</td></tr>";
        }
    }
    $list.="</table>";
    return $list;                           //返回数据数组
}
```

【代码解读】

上面两个函数的本质是对数据库的遍历，属于基础的操作。这里读者需要体会递归方式的使用。

递归的定义是程序调用自身的编程技巧。

具体的描述是一个过程或函数在其定义或说明中又直接或间接调用自身的一种方法，它通常把一个大型复杂的问题，层层转化为一个与原问题相似的规模较小的问题来求解。递归策略只需少量的程序，就可描述出解题过程所需要的多次重复计算，大大地减少了程序的代码量。递归的能力在于，用有限的语句来定义对象的无限集合。一般来说，递归需要有边界条件、递归前进段和递归返回段。当边界条件不满足时，递归前进；当边界条件满足时，递归返回。

注意：递归就是在过程或函数里调用自身。在使用递归策略时，必须有一个明确的递归结束条件，称为递归出口。

HTML 页面（rss_catalog.tpl）的设计如下：

```
栏目:
<label id="lab"></label>
<hr width="97%">
<div align="center">
<a id="hadd" title="添加 RSS 子栏目" class="link"target="_top">添加 RSS 子栏目</a>
<a id="hdel" title="删除" class="link" target="_top">删除</a>
<a id="hmod" title="修改" class="link" target="_top">修改</a>
</div>
<hr width="97%">
<div id="treeDiv">{|$list|}</div>
<script language="javascript" type="text/javascript">
document.getElementById('treeDiv').style.height=(parent.document.bod y.
clientHeight-180);
    findObj('cataml').style.backgroundImage='url(images/open.gif)';
</script>
```

在浏览器中执行下面的链接：

```
http://localhost/lb/admin/rss_catalog.php
```

RSS 栏目树显示的浏览效果如图 18.2 所示。

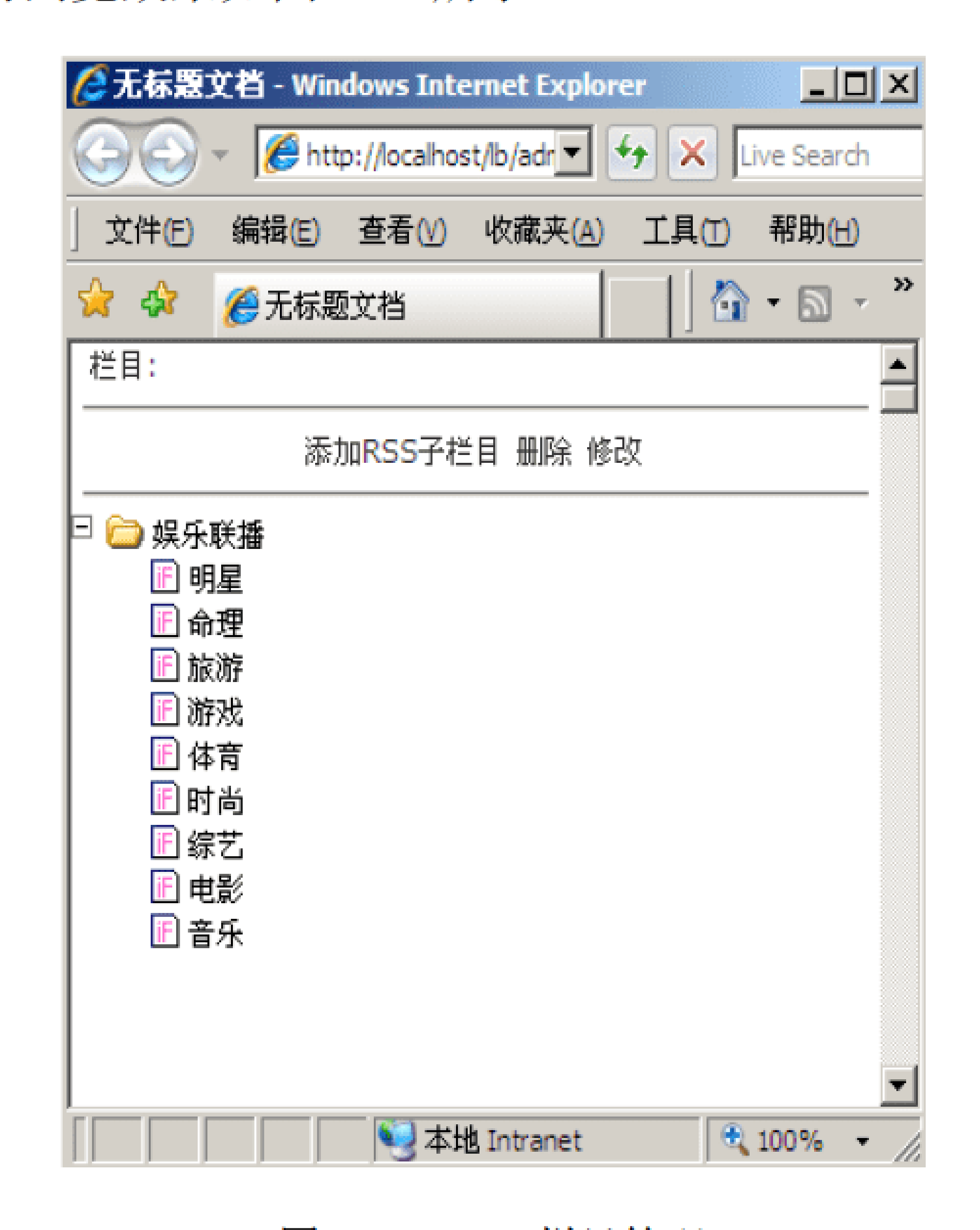


图 18.2　RSS 栏目管理

2. 子栏目添加

子栏目添加功能是根据传入的父 id 创建一个有从属关系的栏目，并通过预先约定的方式写入数据库。创建一个用来显示栏目树的 PHP 脚本文件，命名为 add_rss_catalog_form.php。该文件代码如下：

```
<?php
    //详细内容：根据传入的父 id 值显示新建频道的表单
    //加载配置函数
    include_once( dirname( __FILE__ )."/include.smarty.php");
    include_once( dirname( __FILE__ )."/include.rssfuns.php");
    //变量设置
    $parent_id=var_process("get","parentid");       //从上页得到父 id
    //有效性设置
    if($parent_id=="")
    {
        echo "传入的数据不完整！";
        exit();
        header("location: error.htm");               //重定向浏览器错误页面
    }
    //Smarty 标签替换
    $smarty -> assign("parent_id",$parent_id);      //父 id 标签
    $smarty -> display("./add_rss_catalog_form.tpl");  //加载页面模板
?>
```

【代码解读】

上面的代码中，使用了 var_process()函数对传入父 id 做过滤，这在实际应用中是十分必要的，可以有效避免误操作造成的麻烦。

```
//变量设置
$parent_id=var_process("get","parentid");           //通过 get 的方式得到父 id
```

其中，第一个参数为获得方式，第二个参数是变量的名称。上述代码对应的 HTML 页面模板（add_rss_catalog_form.tpl）的设计如下：

```
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td class="main">
        <form method="post" action="add_rss_catalog.php">
        <table width="100%" border="0" cellspacing="0" cellpadding="0">
            <tr>
                <th colspan="2">//添加子栏目 //</th>
            </tr>
            <tr>
                <td class="Tit">分类名:</td>
                <td><input type="text" name="cataname">
                    <input name="parentid" type="hidden" value="{|$parent_id|}">
                </td>
            </tr>
            <tr>
                <td class="Tit">关键字:</td>
                <td><textarea name="keywordname"></textarea></td>
            </tr>
            <tr>
                <td class="Tit">链接:</td>
```

```
                <td><input type="text" name="urlname"></td>
            </tr>
            <tr>
                <td class="Tit">状态:</td>
                <td><input name="catastatus" type="text" id="catastatus"
                    value="0"></td>
            </tr>
            <tr>
                <td colspan="2" align="center">
                <input name="submit" type="submit" value="提交">
                <input name="reset" type="reset" value="重置">
                </td>
            </tr>
        </table>
        </form>
        </td>
    </tr>
</table>
```

在浏览器中执行下面的链接：

```
http://localhost/lb/admin/add_rss_catalog_form.php?parentid=2
```

栏目管理中的添加子栏目的浏览效果，如图 18.3 所示。

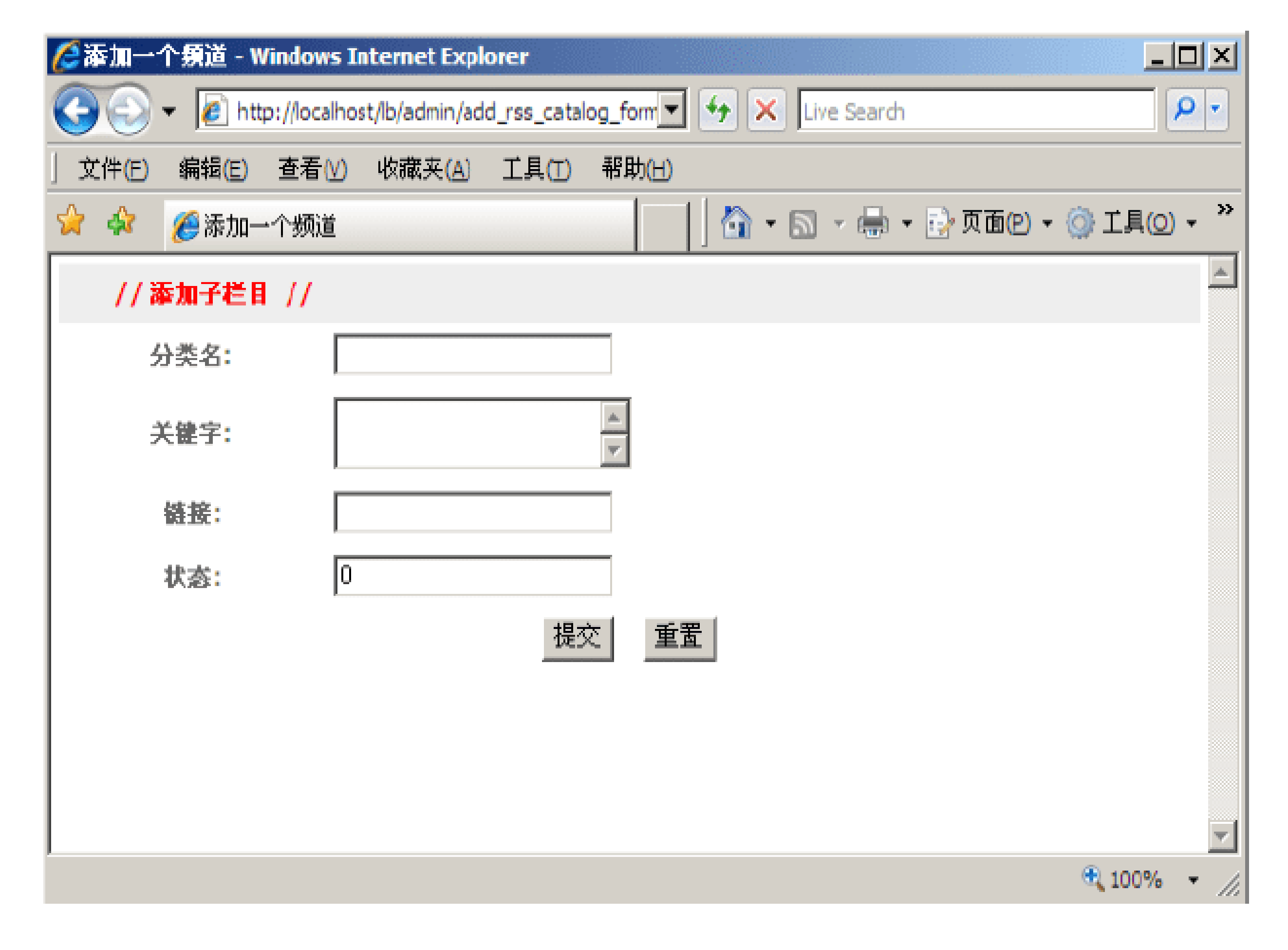


图 18.3　添加子栏目

填写子栏目表单后提交给处理程序，按照规则写入数据库。创建一个用来写入新增栏目的 PHP 脚本文件，命名为 add_rss_catalog.php，该文件代码如下：

```
<?php
    //详细内容：根据表单传入的值，在 newcatalog 表中插入新的数据
    include_once( dirname(__FILE__)."/include.mysqlconfig.php");
    include_once( dirname(__FILE__)."/include.rssfuns.php");
    //变量设置
    $cataname=var_process("post","cataname");           //从表单中得到分类名
    //隐藏变量
    $parentid=var_process("post","parentid");           //从表单中得到父 id
    $keywordname=var_process("post","keywordname");     //获得关键字名称
    $urlname=var_process("post","urlname");             //获得 url 名称
    $catastatus=var_process("post","catastatus");       //获得数据变量
```

```
    //登录状态验证（可以根据实际情况使用 Session 变量实现）
    $login='already';
    if (!empty($login))
    {
        if($cataname=="" || $parentid=="" || $catastatus=="")
                                                    //校验参数是否完整
        {
            echo "传入的参数不完整！";
            exit();
            header("location: error.htm");      //浏览器定向到异常错误页面
        }
        else
        {
    //数据库连接
    $mysql=mysql_connect($mysqlconf["host"],$mysqlconf["user"],$mysqlconf
    ["pass"]);
            mysql_select_db($mysqlconf["cmsdb"]);//选择指定的数据库表（cmsdb）
            $sql = "insert into rss_catalog (cataname,parentid,keywordname,
            urlname,cata_status) values ('".$cataname."',".$parentid.",'".
            $keywordname ."','".$urlname."',".$catastatus.")";
            $result=mysql_query($sql,$mysql);
            mysql_close($mysql);                //关闭数据库连接
            if($result)                         //判断结果集是否存在
            {
                echo "添加成功！";
                exit();
                header("location: success.htm");//浏览器定向到操作成功页面
            }
            else
            {
                echo "添加失败！";
                exit();
                header("location: error.htm"); //浏览器定向到异常错误页面
            }
        }
    }
    else
    {
        echo "<script language=\"javascript\" type=\"text/javascript\">alert
        (\"您没有权限进行此操作，请与管理员联系！\");history.back();</script>";
        header("Location: index.php?error_msg=您没有权限操作" );
                                                    //浏览器定向到异常错误页面
        exit();
    }
?>
```

填入正确的栏目信息后提交给上面的程序处理，程序的输出如下：

```
添加成功！
```

3. 栏目修改

栏目修改程序是获得基本参数信息需要的工具，其设计目的是提供给管理员修改相应的栏目，它主要的功能是重新编辑已经录入的栏目基本参数信息。创建一个用来修改栏目基本参数信息的 PHP 脚本文件，命名为 modify_rss_catalog_form.php，该文件代码如下：

```
<?php
    //详细内容：显示对栏目的内容进行修改的表单，根据传入的频道 id
    //加载配置文件
    include_once( dirname( __FILE__ )."/include.smarty.php");
    include_once( dirname( __FILE__ )."/include.mysqlconfig.php");
    include_once( dirname(__FILE__)."/include.rssfuns.php");
    //变量设置
    $id=var_process("get","id");           //得到频道 id
    //测试数据完整性
    if($id == "")
    {
        echo "传入的数据不完整";
        exit();
    }
    $mysql=mysql_connect($mysqlconf["host"],$mysqlconf["user"],$mysqlconf
    ["pass"]);
    mysql_select_db($mysqlconf["cmsdb"]);
    $sql = "select * from rss_catalog where id=" . $id;
    $result=mysql_query($sql,$mysql);
    if($result==false)
    {
        echo "没有找到这个分类！";
        header("location: error.htm");     //浏览器定向到异常错误页面
        exit();
    }
    $arr=mysql_fetch_array($result);
    mysql_close($mysql);//关闭数据库
    //生成 Smarty 标签
    $smarty -> assign("id",$id);                            //频道 id
    $smarty -> assign("cataname",$arr["cataname"]);         //频道名称
    $smarty -> assign("keywordname",$arr["keywordname"]);   //关键字
    $smarty -> assign("urlname",$arr["urlname"]);           //url 链接
    $smarty -> assign("catastatus",$arr["cata_status"]);    //类型
    //修改页面模板
    $smarty -> display("./modify_rss_catalog_form.tpl");
?>
```

【代码解读】

以上的代码通过链接地址传递的参数值，取出对应栏目分类信息。由于只有一条信息，所以没有使用循环遍历，直接使用 mysql_fetch_array()获得结果数组。代码如下：

```
$arr=mysql_fetch_array($result);
mysql_close($mysql);
```

上述代码对应的 HTML 页面模板代码（modify_rss_catalog_form.tpl）的设计如下：

```
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td class="main">
        <form method="post" action="modify_rss_catalog.php">
        <table width="60%" border="0" cellspacing="1" cellpadding="0">
            <tr>
                <th colspan="2">//修改此分类的基本内容 //</th>
            </tr>
            <tr>
                <td class="Tit">分类名:</td>
                <td><input type="text" name="cataname" value="{|$cataname|}">
```

```
                <input type="hidden" name="id" value="{|$id|}"></td>
            </tr>
            <tr>
                <td class="Tit">关键字:</td>
                <td><textarea name="keywordname">{|$keywordname|}</textarea></td>
            </tr>
            <tr>
                <td class="Tit">链接:</td>
                <td><input type="text" name="urlname" value="{|$urlname|}"></td>
            </tr>
            <tr>
                <td class="Tit">状态:</td>
                <td><input name="catastatus" type="text" id="catastatus"
                    value="{|$catastatus|}"></td>
            </tr>
            <tr>
                <td colspan="2" align="center"><input name="submit"
                    type="submit" value="提交">    <input name="reset"
                    type="reset" value="重置"></td>
            </tr>
        </table>
        </form>
        </td>
    </tr>
</table>
```

在栏目显示页面选择修改之前录入的“紫薇斗数”栏目，或者在浏览器中执行下面的链接：

```
http://localhost/lb/admin/modify_rss_catalog_form.php?id=18
```

栏目修改界面的浏览效果，如图 18.4 所示。

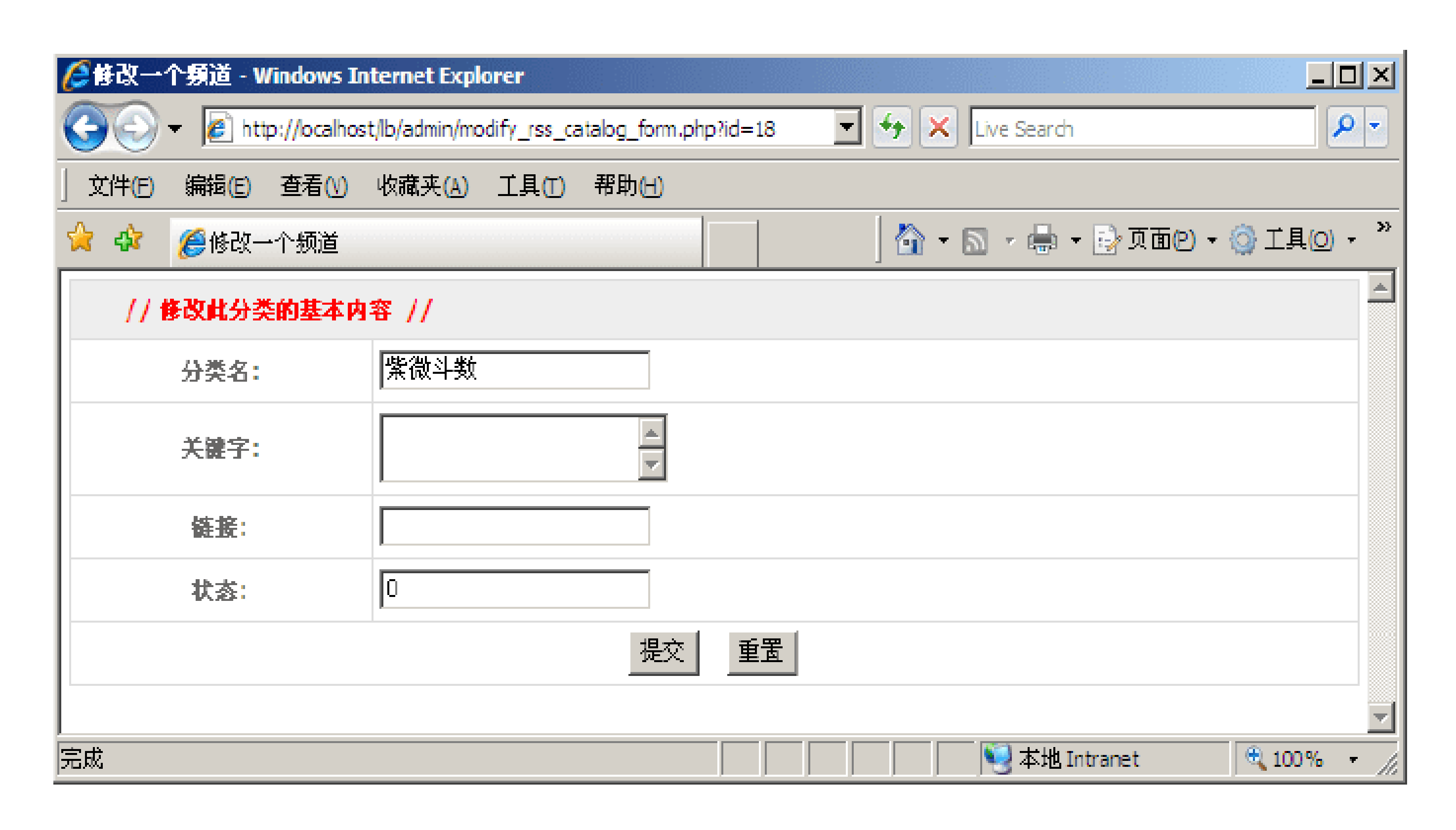


图 18.4　修改栏目

修改完毕后，提交给栏目内容修改程序（modify_rss_catalog.php），处理后更新数据库中对应的内容，该文件代码如下：

```
<?php
    //详细内容：对栏目的内容进行修改根据 id
    //加载配置文件
    //include_once( dirname( __FILE__ )."/include.session.php" );
    include_once( dirname( __FILE__ )."/include.mysqlconfig.php");
    include_once( dirname(__FILE__)."/include.rssfuns.php");
```

```
//登录状态验证（可以根据实际情况使用 Session 变量实现）
$login='already';
if (!empty($login))                              //判断是否登录
{
    //变量设置
    $id=var process("post","id");                //隐藏变量
    $cataname=var_process("post","cataname");    //从表单中得到要插入的值
    $keywordname=var process("post","keywordname");
    $urlname=var process("post","urlname");
    $catastatus=var process("post","catastatus");

    //有效性检查
    if( $id=="" || $cataname=="" || $catastatus=="")
    {
        echo "传入的参数不完整！";
        exit();

        //header("location: error.htm");   //浏览器定向到异常错误页面
    }
    else
    {

$mysql=mysql connect($mysqlconf["host"],$mysqlconf["user"],$mysqlconf
["pass"]);
        mysql select db($mysqlconf["cmsdb"]);

        $sql = "update rss catalog set cataname='". $cataname ."',
        keywordname='". $keywordname ."',urlname='". $urlname ."',
        cata status='". $catastatus ."' where id=" . $id;
        $result=mysql query($sql,$mysql);
        mysql close($mysql);            //关闭数据库
        if($result)
        {
            echo "修改成功！";
            exit();
            //header("location: success.htm");//浏览器定向到修改成功提示页面
        }
        else
        {
            echo "修改失败！";
            exit();
            //header("location: error.htm");//浏览器定向到异常错误页面
        }
    }
    }
    else
    {
        echo "<script language=\"javascript\" type=\"text/javascript
       \">alert(\"您没有权限进行此操作，请与管理员联系！\");history.back();
        </script>";
    header("Location: index.php?error_msg=您没有权限操作" );
                                            //浏览器定向到异常状态
```

```
        exit();
    }
?>
```

提交后程序输出如下内容：

```
修改成功!
```

4．栏目删除

栏目修改程序也是基础的管理工具，设计目的是提供给管理员删除指定栏目。程序的执行流程分为两步骤实现：

（1）询问是否删除栏目。

（2）当得到确认之后，在数据库中删除指定栏目信息。

创建一个用来确认删除栏目的 PHP 脚本文件，命名为 del_rss_catalog_form.php。该文件代码如下：

```
<?php
    //详细内容：用户提出删除操作时判断是否可以删除，弹出的提示
    //加载配置文件
    include once( dirname(  FILE  )."/include.mysqlconfig.php");
    include once( dirname(  FILE  )."/include.rssfuns.php");
    include once( dirname(  FILE  )."/include.smarty.php");
    $clew="";
    //变量设置
    $id=var process("get","id");
    //判断 id 是否为空（异常提交）
    if($id == "")
    {
        $clew = "传入的数据不完整！";
    }
    else
    {
        $mysql=mysql connect($mysqlconf["host"],$mysqlconf["user"],
        $mysqlconf["pass"]);
        mysql select db($mysqlconf["cmsdb"]);
        //SQL 查询
        $sql = "select * from rss_catalog where parentid=" . $id;
        $result=mysql query($sql,$mysql);
        $num=mysql num rows($result);
        if($num)                    //判断本栏目下存在子栏目
        {
            $clew="警告!!!本栏目下存在子栏目!您不能删除!";
        }
        else
        {
            //判断目录下是否有相关 RSS 源
            $sql = "select * from rss_source where catalogid=" . $id;
            $result=mysql query($sql,$mysql);
            $num=mysql num rows($result);
            if($num)                //有文章则跳转
```

```
                {
                    $clew="警告!!!本栏目下存在相关 RSS 源!您不能删除!";
                }
                else
                {
                    $clew="是否删除本栏目?  <a href=\"del rss catalog.
                    php?id=". $id ."\">确认</a>";
                }
            }
        }
        //smarty 标签替换
        $smarty -> assign("clew",$clew);
        $smarty -> display("./del_rss_catalog_form.tpl");  //加载删除页面模板
?>
```

在浏览器中执行下面的链接：

```
http://localhost/lb/admin/del_rss_catalog_form.php?id=18
```

确认删除栏目的界面程序的浏览效果，如图 18.5 所示。

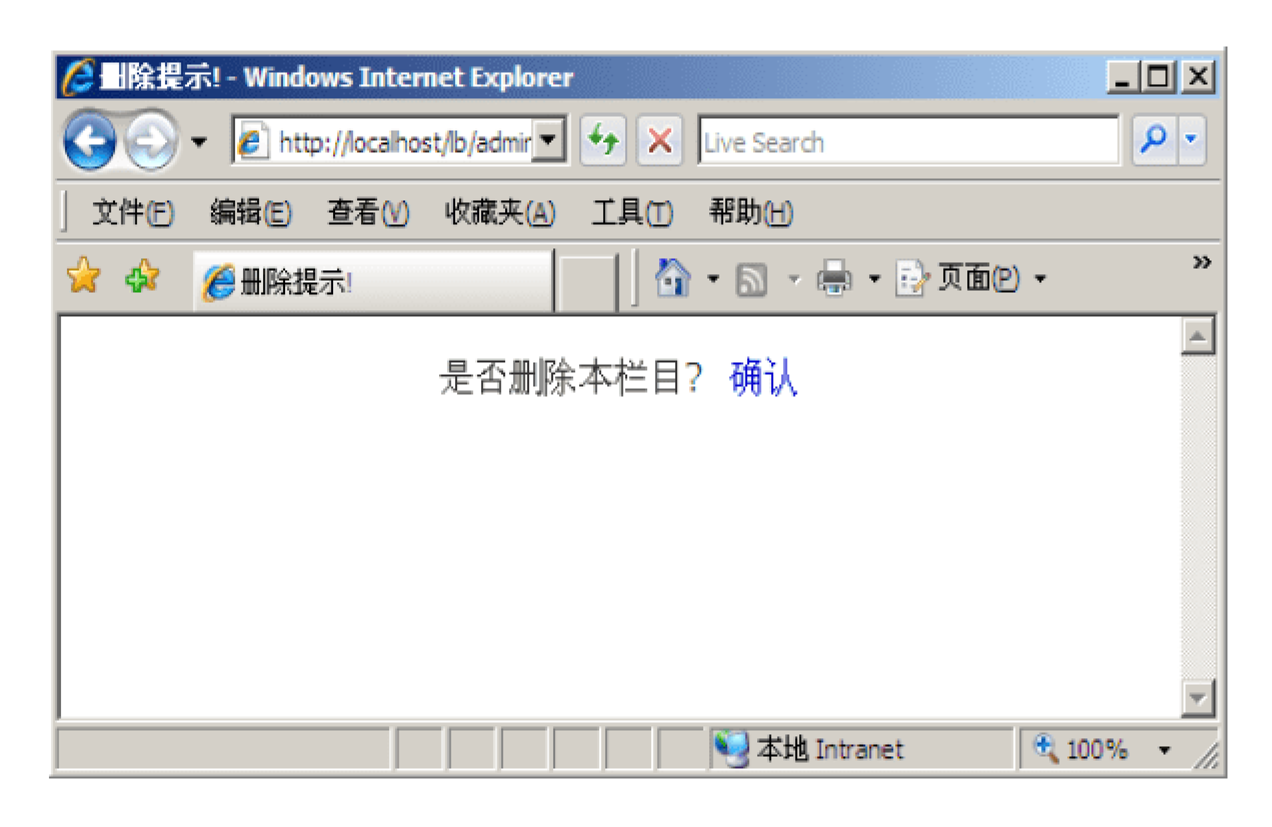


图 18.5　栏目删除确认

创建从数据库中删除指定栏目信息的 PHP 脚本文件，命名为 del_rss_catalog_form.php。代码如下：

```
<?php
    //详细内容：删除一个频道，根据传入的频道 id
    //加载配置文件
    //include once( dirname(  FILE  )."/include.session.php" );
    include once( dirname(  FILE  )."/include.mysqlconfig.php");
    include once( dirname(  FILE  )."/include.rssfuns.php");
    //变量设置
    $id=var process("get","id");
    //登录状态验证（可以根据实际情况使用 Session 变量实现）
    $login='already';
    if (!empty($login))                          //检查登录状态
    {
        //有效性检查
        if($id=="")
        {
```

```
            echo "传入的参数不完整！";
            exit();
            //header("location: error.htm");   //浏览器定向到异常错误页面
        }
        //判断本栏目下存在子栏目
        $mysql=mysql connect($mysqlconf["host"],$mysqlconf["user"],
        $mysqlconf["pass"]);
        mysql select db($mysqlconf["cmsdb"]);
        $sql = "select * from rss_catalog where parentid=" . $id;
        $result=mysql query($sql,$mysql);
        $num=mysql num rows($result);
        if($num>0)
        {
            header("location: error.htm");     //浏览器重定向
        }
        //判断目录下是否有相关文章
        $sql = "select * from rss_source where catalogid=" . $id;
        $result=mysql query($sql,$mysql);
        $num=mysql_num_rows($result);          //获得数据集的行数
        if($num)
        {
            echo "存在相应的 RSS，不允许删除！";
            exit();
            header("location: error.htm");     //浏览器重定向错误页面
        }
        //删除该栏目
        $sql="delete from rss catalog where id=" . $id;
        $result=mysql query($sql,$mysql);
        if($result)
        {
            echo "删除成功！";
            exit();
            header("location: success.htm");   //浏览器重定向错误页面
        }
        Else                                   //删除操作异常处理
        {
            echo "删除失败！";
            exit();
            header("location: error.htm");     //浏览器重定向错误页面
        }
        mysql_close($mysql);// 关闭数据库

        }
        else
        {
            echo "<script language=\"javascript\" type=\"text/javascript\
            ">alert(\"您没有权限进行此操作，请与管理员联系！\");history.back();
            </script>";
            header("Location: index.php?error msg=您没有权限操作" );
            exit();
    }
?>
```

【代码解读】

该程序的执行流程分为 3 个步骤：

（1）对传入参数进行验证。

（2）判断该栏目下面是否有已经抓取的文章或者是否有子栏目。

（3）再执行数据库的删除操作。

以上提到的这 3 个步骤在代码实现上都比较容易，读者也可以很容易看懂。这里需要注意理解业务的执行流程和步骤。

确认删除后程序的输出如下提示信息：

```
删除成功!
```

18.4.2　添加 RSS 源

添加 RSS 源是在之前栏目的基础上设置 RSS 的抓取源的基本信息，需要设定的信息包括所属栏目、RSS 来源、RSS 地址、抓取间隔时间、RSS 抓取状态、编辑 6 个部分。其中，前 5 项为抓取时的必要参数，这部分的程序执行流程分为两个部分，即显示添加表单程序和添加入库程序。

（1）创建一个用来显示 RSS 源表单的 PHP 脚本文件，命名为 new_rss_source.php。该文件代码如下：

```
<?php
//详细介绍：增加 RSS
include_once (dirname ( __FILE__ ) . "/include.smarty.php");//取得 smarty 类
include_once (dirname ( __FILE__ ) . "/include.mysqlconfig.php");
                                                    //取得 mysql 参数
include_once (dirname ( __FILE__ ) . "/include.rssfuns.php");
                                                    //取得 RSS 的函数库
//创建数据库链接
$mysql = mysql_connect ( $mysqlconf ["host"], $mysqlconf ["user"],
$mysqlconf ["pass"] );
mysql_select_db ( $mysqlconf ["cmsdb"] );
mysql_query ( "SET NAMES utf-8" );
$cataid_list = "1";                                 //设定列表父类
$son_cataid = get_ids ( $cataid_list, $mysql );     //取得该父类下的子类
$rss_catalog = sel_catalog ( $son_cataid, $mysql );//取得所有分类的详细信息
$rss_cata_nums = count ( $rss_catalog );            //分类栏目下拉菜单开始
$cata_list = "";
for($i = 0; $i < $rss_cata_nums; $i ++) {
    if ($i == 0)                                    //默认第一个显示的
{
    $cata_list .= "<option value=\"" . $rss_catalog [$i] ["id"] . "\" selected>".
    $rss_catalog [$i] ["cataname"] . "</option>";
    } else {
        $cata_list .= "<option value=\"" . $rss_catalog [$i] ["id"] . "\" >" .
        $rss_catalog [$i] ["cataname"] . "</option>";
    }
}
$smarty->assign ( "cata_list", $cata_list );        //数据列表
$smarty->assign ( "editor_name", $_SESSION ["username"] );
//分类栏目下拉菜单结束
$smarty->display ( "./new_rss_source.tpl" );
mysql_close ( $mysql );
?>
```

【代码解读】

上面的代码中，需要注意设定初始的父 id，这样才能使用 get_ids()函数正确地获得全部子栏目 id。代码如下：

```
//设定列表父类
$cataid_list="1";
//取得该父类下的子类
$son_cataid=get_ids($cataid_list,$mysql);
```

（2）使用 sel_catalog()函数获得详细的分类信息，代码如下：

```
//取得所有分类的详细信息
$rss_catalog=sel_catalog($son_cataid,$mysql);
```

以上代码对应的 HTML 页面模板（new_rss_source.tpl）的设计如下：

```
<table width="100%" border="0" cellpadding="0" cellspacing="0"
    id="mastPath">
    <tr>
        <td>您的位置：<a href="loginindex.php">首页</a> &raquo; <a
            href="article_list.php">娱乐联播</a> &raquo; 添加 RSS 源</td>
    </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td valign="top" class="Left">
        <table width="100%" border="0" cellspacing="0" cellpadding="0">
            <tr>
                <td>&oplus; <span class="Boo"><a href="rss_catalog.php">
                RSS 栏目管理</a></span></td>
            </tr>
            <tr>
                <td>&oplus; <a href="new_rss_source.php">添加 RSS 源</a></td>
            </tr>
            <tr>
                <td>&oplus; <a href="rss_source_list.php">已有 RSS 源管理
                </a></td>
            </tr>
            <tr>
                <td>&oplus; <a href="rss_content_list.php">RSS 头条管理
                </a></td>
            </tr>
        </table>
        </td>
        <td class="main">
        <form action="add_rss_source.php" method="post" name="form1"
            onsubmit="return check();">
        <table width="100%" border="0" align="center" cellpadding="0"
            cellspacing="1">
            <tr>
                <th colspan="2">// 添加 RSS 源 //</th>
            </tr>
            <tr>
                <td width="15%" align="center" class="Tit">所属栏目
                <td width="85%"><select name="catalogid">
                    {|$cata_list|}
                </select></td>
            </tr>
```

```
            <tr>
                <td align="center" class="Tit">RSS 来源</td>
                <td><input name="source" type="text" id="title" size="80">
                </td>
            </tr>
            <tr>
                <td align="center" class="Tit">RSS 地址</td>
                <td><input name="rss_url" type="text" size="80" /></td>
            </tr>
            <tr>
                <td align="center" class="Tit">抓取间隔时间</td>
                <td><input name="get_time" type="text" size="10"
                    maxlength="200" value="{|$get_time|}"></td>
            </tr>
            <tr>
                <td align="center" class="Tit">RSS 抓取状态</td>
                <td><input name="rss_status" type="radio"value="1" checked />
                开启抓取 <input name="rss_status" type="radio" value="0" />
                关闭抓取</td>
            </tr>
            <tr>
                <td align="center" class="Tit">编辑<br>
                </td>
                <td><input name="author" type="text" size="10"maxlength="200"
                    value="{|$editor_name|}" readonly="true"></td>
            </tr>
            <tr>
                <td colspan="2" class="Centre"><input type="submit"
                    name="submit" value=" 新 增 "></td>
            </tr>
        </table>
        </form>
        </td>
    </tr>
</table>
```

在浏览器中执行下面的链接：

```
http://localhost/lb/admin/new_rss_source.php
```

添加 RSS 源界面程序的浏览效果，如图 18.6 所示。

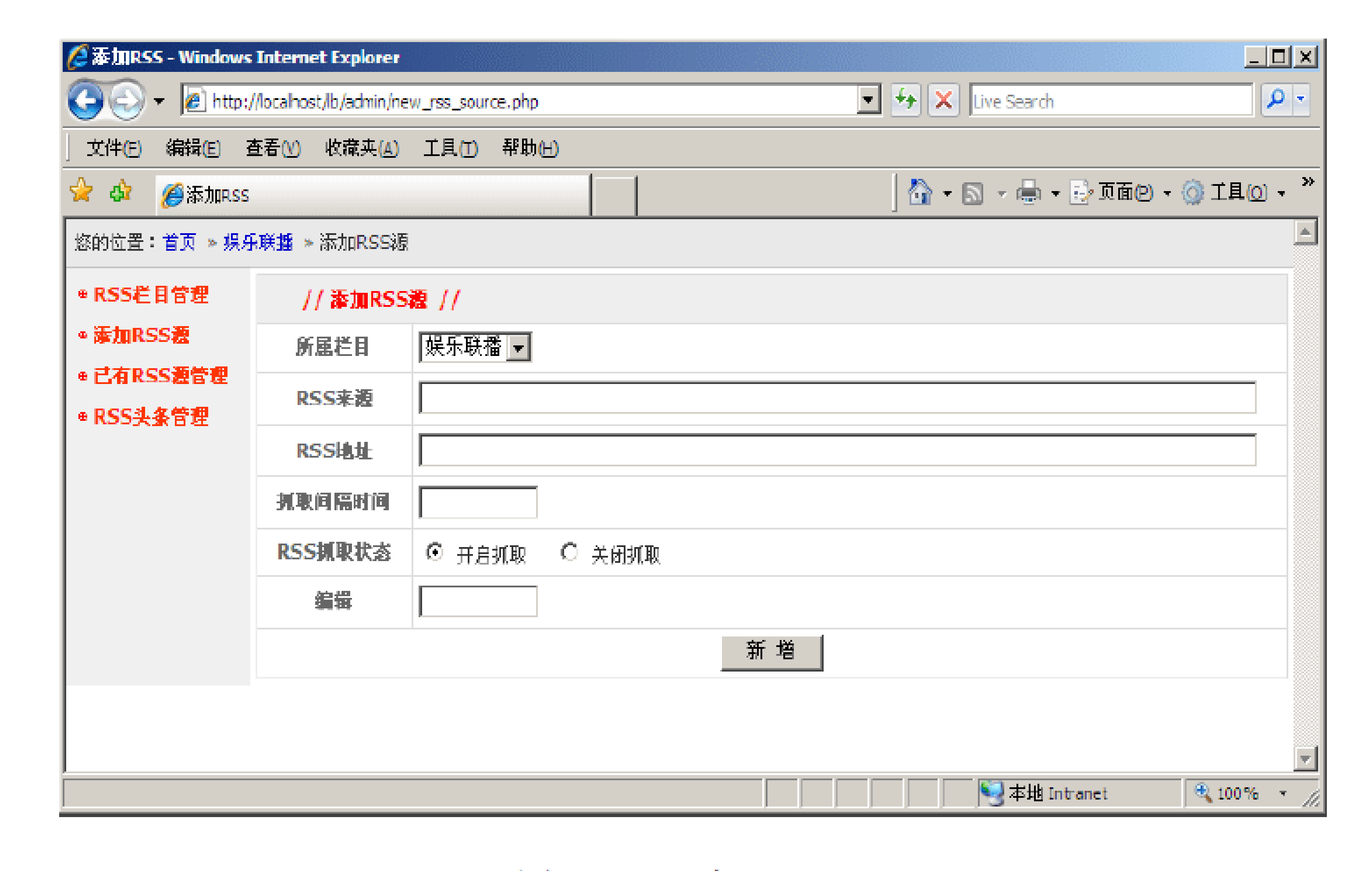


图 18.6　添加 RSS 源

下面创建用来接收 RSS 源表单信息，并将整理入库的 PHP 脚本文件，命名为 add_rss_source.php。该文件代码如下：

```
<?php
//详细介绍：添加新的 RSS 源
include_once (dirname ( __FILE__ ) . "/include.mysqlconfig.php");
                                                    //取得 mysql 参数
include_once (dirname ( __FILE__ ) . "/include.rssfuns.php");
                                                    //取得 rss 的函数库
//创建数据库连接
$mysql = mysql_connect ( $mysqlconf ["host"], $mysqlconf ["user"], $mysqlconf
["pass"] );
mysql_select_db ( $mysqlconf ["cmsdb"] );           //选择指定表（cmsdb）
//安全接受参数信息
$catalogid = var_process ( "post", "catalogid" );
$rss_url = var_process ( "post", "rss_url" );
$source = var_process ( "post", "source" );
$rss_status = var_process ( "post", "rss_status" );
$get_time = var_process ( "post", "get_time" );
$author = var_process ( "post", "author" );
//异常验证
if ($catalogid == "" or $rss_url == "" or $source == "" or $get_time == ""
or $rss_status == "" or $author == "") {
    echo "数据传输错误,请重试  ";
    echo "<a href=\"new_rss_source.php\">返回重试</a>";
    exit ();
}
//判断 RSS 源是否已经存在
if (RssSourceExist ( $rss_url, $mysql )) {
    echo "<script language=\"javascript\" type=\"text/javascript\">alert
    (\"这个 RSS 源网址已经存在了！ \");history.back();</script>";
    exit ();
}
//插入 RSS 源信息
$flag = ins_source ( $catalogid, $rss_url, $source, $get_time, $author,
$rss_status, $mysql );
if ($flag) {
    echo "RSS 源添加成功  ";
    echo "<a href=\"rss_source_list.php\">查看源列表</a>";
    exit ();
} else {
    echo "RSS 源添加失败,请重试  ";
    echo "<a href=\"new_rss_source.php\">返回重试</a>";
    exit ();
}
mysql_close ( $mysql );     //关闭数据库
?>
```

【代码解读】

以上的代码首先要排除参数的异常情况（安全接收参数），然后是使用 RssSourceExist()判断 RSS 源是否已经存在，再使用 ins_source()函数将 RSS 数据源信息写入数据库。

其中，RssSourceExist()会判断 RSS 源的个数，如果查询失败返回 0。具体的函数代码如下：

```
//判断 RSS 源的个数
function RssSourceExist($rss_url,$mysql)
```

```
{
    $sql="select count(*) from rss_source where rss_url='". $rss_url ."'";
    $result=mysql_query($sql,$mysql);
    if($result)                                  //判断是否获得结果集
    {
        $arr=@mysql_fetch_array($result);        //将查询结果赋值给变量
        return $arr[0];
    }
    return 0;
}
```

ins_source()函数是封装对数据库的写入操作，并返回写入操作的结果。具体的函数代码如下：

```
//添加 RSS 源（封装对数据库的写入操作）
function ins_source($catalogid, $rss_url,$source,$get_time,$author, $rss_status,$mysql)
{
    $sql="insert into rss_source (catalogid,rss_url,source,get_time,author,
    update_time,rss_status)"
."values('".$catalogid."','".$rss_url."','".$source."','".$get_time."',
'".$author."',NOW(),'".$rss_status."')";
    $result=mysql_query($sql,$mysql);
    return $result ;        //返回状态
}
```

18.4.3　已有 RSS 源管理

RSS 源的管理部分提供管理编辑和删除这两个基本功能。读者在实际操作中，可以根据网站的情况调整 RSS 源的参数设置。创建一个用来显示 RSS 源表单的 PHP 脚本文件，命名为 rss_source_list.php，该文件代码如下：

```
<?php
    //详细介绍：RSS 源列表
    include_once( dirname( __FILE__ )."/include.smarty.php" );
                                                //取得 smarty 类
    include_once( dirname( __FILE__ )."/include.mysqlconfig.php" );
                                                //取得 mysql 参数
    include_once( dirname( __FILE__ )."/include.rssfuns.php" );
                                                //取得 rss 的函数库
    // 打开数据库
    $mysql = mysql_connect( $mysqlconf[ "host" ], $mysqlconf[ "user" ],
    $mysqlconf[ "pass" ] );
    mysql_select_db( $mysqlconf[ "cmsdb" ] );
    $cataid_list="1";//设定列表父类
    $son_cataid=get_ids($cataid_list,$mysql);      //取得该父类下的所有子类
    $rss_catalog=sel_catalog($son_cataid,$mysql);  //取得所有分类的详细信息
    //取得下拉菜单的被选中项 id，如果没有则默认为娱乐联播
    $rss_cataid=var_process("get","rss_cataid",1);
    $rss_cata_nums=count($rss_catalog);           //计算数量
    $cata_list="";
    for($i=0;$i< $rss_cata_nums;$i++)
    {
        //如果该选项跟被选中项目相同，则设定其为被选中
        if($rss_cataid==$rss_catalog[$i]["id"])
```

```
        {
            $cata list .="<option value=\"".$rss catalog[$i]["id"]."\"
            selected>".$rss catalog[$i]["cataname"]."</option>";
        }
        Else                                    //如果没有被选中生成样式
        {
            $cata list .="<option value=\"".$rss catalog[$i]["id"]."\"
            >".$rss catalog[$i]["cataname"]."</option>";
        }
    }
    $smarty->assign("cata_list",$cata_list);   //生成列表模板标签
```

以上代码设置获得分类的参数、遍历数据库中的 RSS 源列表信息，将获得的结果集生成对应的 Smarty 模板标签。下面介绍如何将获得信息以列表的形式展示（输出到浏览器中），代码如下：

```
//分类 rss 源列表开始
//列表分类开始
$son_cataid=get_ids($rss_cataid,$mysql); //取得下拉菜单中被选中项的子 ID
$source_nums = source_nums($son_cataid,$mysql);    //获得分类文章总数
$currpage = var_process("get","page",1);           //获得参数当前页数
$itemnum = 20;                                     //单页条目数
$listpage = divpage toadmin( $source nums, $currpage, $itemnum );
                                                   //获得页码
$list_offset = $listpage["rec_start"];             //偏移值
$firstpage = "<a href=\"rss source list.php?rss cataid=".$rss cataid.
"&&page=".$listpage["firstpage"]."\">首页</a>";
$prepage = "<a href=\"rss source list.php?rss cataid=".$rss cataid.
"&&page=".$listpage["prepage"]."\">上一页</a>";
$nextpage = "<a href=\"rss source list.php?rss cataid=".$rss cataid.
"&&page=".$listpage["nextpage"]."\">下一页</a>";
$lastpage = "<a href=\"rss_source_list.php?rss_cataid=".$rss_cataid.
"&&page=".$listpage["max_page"]."\">尾页</a>";
$smarty->assign("rss_source_nums",$source_nums);      //总条数标签
$smarty->assign("page nums",$listpage["max page"]);   //生成总页数标签
$smarty->assign("firstpage",$firstpage);              //首页标签
$smarty->assign("lastpage",$lastpage);                //尾页标签
$smarty->assign("prepage",$prepage);                  //上一页标签
$smarty->assign("nextpage",$nextpage);                //下一页标签
$smarty->assign("now page",$currpage);                //当前页标签
//列表分类结束
//取指定条目的 rss 源列表开始
//取得该子 id 下的所有 rss 源
$rss source = source list( $son cataid,$list offset,$itemnum,$mysql);
$rss_source_list_nums=count($rss_source);  //计算总 rss 源共有多少条
$rss source list="";
//for 循环列出所有的 rss 源
for($i=0;$i< $rss source list nums;$i++)
{
$rss source list.="<tr><td>".$rss source[$i]["id"]."</td><td>".$rss
source[$i]["source"]."</td><td>".$rss source[$i]["rss url"]."</td><td>"
."<a href=\"edit rss source.php?id=".$rss source[$i]["id"]."\"onClick=
'return confirm(\" 你真的要编辑该 RSS 吗？ \");'>编辑</a> | "
```

```
    ."<a href=\"del rss source.php?id=".$rss source[$i]["id"]."\" onClick=
    'return confirm(\" 确定要删除吗？ \");'>删除</a></td></tr>";
    }
    $smarty->assign("rss_source_list",$rss_source_list);      //列表标签
    //取指定条目的 rss 源列表结束
    //加载模板
    $smarty->display("./rss source list.tpl");
    mysql close($mysql);
?>
```

【代码解读】

由于 RSS 源管理的代码是通用程序，所以首先需要获得栏目的分类 id，具体处理的代码如下：

```
$cataid list="1";
$son_cataid=get_ids($cataid_list,$mysql);      //取得该父类下的所有子类
$rss_catalog=sel_catalog($son_cataid,$mysql);  //取得所有分类的详细信息
//取得下拉菜单的被选中项 id，如果没有，则默认为娱乐联播
$rss_cataid=var_process("get","rss_cataid",1);
```

以上的代码已经获得了 RSS 源管理列表数据数组，以这些 id 为条件将对应的数据从数据库中遍历出来，并生成模板标签输出到模板中，完成了 RSS 源管理列表的功能。对应的 HTML 页面模板代码（rss_source_list.tpl）的设计如下：

```
<table width="100%" border="0" cellpadding="0" cellspacing="0"
    id="mastPath">
    <tr>
        <td>您的位置：<a href="loginindex.php">首页</a> &raquo; <a
            href="article_list.php">娱乐联播</a> &raquo; RSS 源列表管理</td>
    </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td valign="top" class="Left">
        <table width="100%" border="0" cellspacing="0" cellpadding="0">
            <tr>
                <td>&oplus; <span class="Boo"><a href="rss catalog.php">
                RSS 栏目管理</a></span>
                </td>
            </tr>
            <tr>
                <td>&oplus; <a href="new_rss_source.php">添加 RSS 源</a></td>
            </tr>
            <tr>
                <td>&oplus; <a href="rss_source_list.php">已有 RSS 源管理</a></td>
            </tr>
            <tr>
                <td>&oplus; <a href="rss content list.php">RSS 头条管理</a></td>
            </tr>
        </table>
        </td>
        <td class="main">
```

```
        <form name="form2" method="get" action="rss source list.php">
        <table width="100%" border="0" align="center" cellpadding="0"
            cellspacing="0" class="mainHead">
            <tr>
                <td width="30%">// RSS 源管理 //</td>
                <td width="70%" align="right">栏目选择： <select
                name="rss cataid"
                    onChange="submit();">
                    {|$cata list|}
                </select></td>
            </tr>
        </table>
        </form>
        <table width="100%" border="0" cellpadding="0" cellspacing="1"
            class="TableCenter">
            <tr class="Tit">
                <td width="12%">RSS 编号</td>
                <td width="13%">RSS 源名称</td>
                <td width="60%">RSS 源网址</td>
                <td width="15%">操作</td>
            </tr>
            {|$rss source list|}
        </table>
        <table width="100%" height="30" border="0" cellpadding="0"
            cellspacing="0">
            <tr>
                <td width="50%">共<span class="Boo">{|$rss source nums|}
                </span>条 rss 记录
                当前显示第<span class="Boo">{|$now_page|}</span>页  
                共<span
                    class="Boo">{|$page nums|}</span>页</td>
                <td width="50%" align="right">{|$firstpage|} | {|$prepage|} |
                {|$nextpage|} | {|$lastpage|} | 到 <input name="currpage"
                type="text"
                    size="3" class="input jump"> 页</td>
                <form name="form1" method="post" action="{|$itempage|}">
                </form>
            </tr>
        </table>
        </td>
    </tr>
</table>
```

在浏览器中执行下面的链接：

```
http://localhost/lb/admin/rss_source_list.php
```

RSS 源列表界面程序的浏览效果如图 18.7 所示。

RSS 管理功能包括编辑和删除两个基本功能。编辑功能的实现原理与添加 RSS 源的原理一致，只是将 insert 入库的操作，变成更新指定数据的操作。请读者根据添加 RSS 源程序，自行修改编辑功能程序，这里不再赘述。删除指定 RSS 源的功能的代码如下：

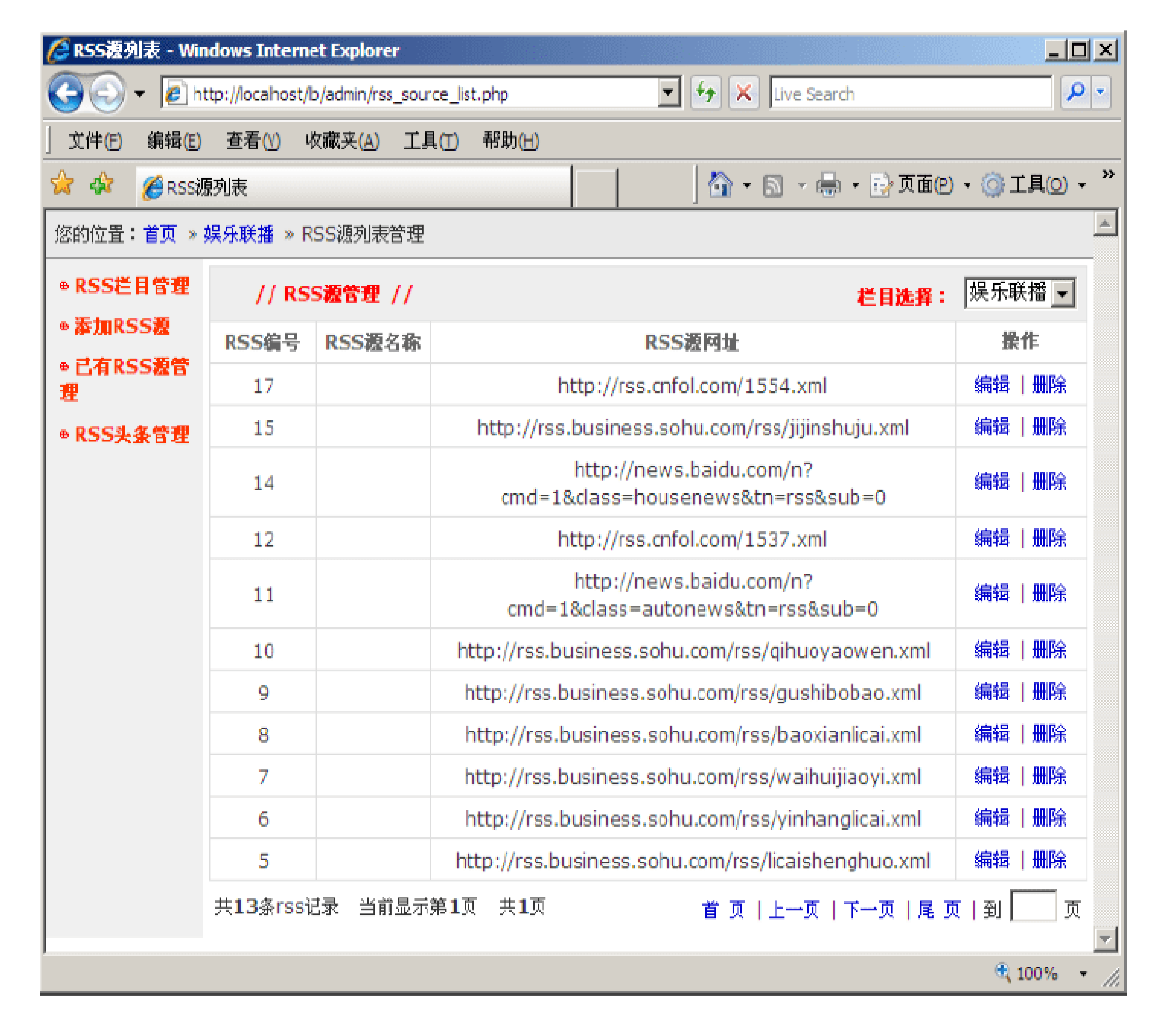


图 18.7　RSS 源管理列表

```
<?php
//详细介绍：删除 rss
include_once (dirname ( __FILE__ ) . "/include.smarty.php");//取得 smarty 类
include once (dirname (  FILE  ) . "/include.mysqlconfig.php");
                                             //取得 mysql 参数
include once (dirname (  FILE  ) . "/include.rssfuns.php");
                                             //取得 RSS 的函数库
//创建数据库连接
$mysql = mysql connect ( $mysqlconf ["host"], $mysqlconf ["user"],$mysqlconf
["pass"] );
mysql select db ( $mysqlconf ["cmsdb"] );
$id = var_process ( "get", "id" );           //获得删除 id
if ($id == "") {                             //判断待删除 id 是否存在
    header ( "Location: rss source list.php" );
    exit ();
} else {                                     //存在则进入删除操作
    $flag = del_source ( $id, $mysql );      //执行删除操作
    if ($flag) {
        echo "RSS 源删除成功";
        echo "<a href=\"rss_source_list.php\">查看源列表</a>";
        exit ();
    } else {
        echo "RSS 源删除失败";
        echo "<a href=\"rss source list.php\">返回重试</a>";
        exit ();
    }
}
mysql close ( $mysql );                      //关闭数据库
?>
```

选择指定的 RSS 源单击“删除”链接，程序输入如下提示信息：

```
RSS 源删除成功
```

18.4.4　RSS 头条管理

RSS 头条管理程序，是对编辑推荐到首页的内容进行管理的程序。推荐的过程是对指定文章的“状态标识位”操作来实现的。

（1）创建一个用来显示 RSS 头条信息的 PHP 脚本文件，命名为 rss_content_list.php。该文件代码如下：

```
<?php
//详细介绍：头条 RSS 源列表
include_once (dirname ( __FILE__ ) . "/include.smarty.php");//取得 smarty 类
include once (dirname (   FILE   ) . "/include.mysqlconfig.php");
                                         //取得 mysql 参数
include_once (dirname ( __FILE__ ) . "/include.rssfuns.php");
                                         //取得 rss 的函数库
//创建数据库连接
$mysql = mysql connect ( $mysqlconf ["host"], $mysqlconf ["user"],$mysqlconf
["pass"] );
mysql select db ( $mysqlconf ["cmsdb"] );
//取指定条目的 rss 源列表开始
$sql = "select id ,title ,source from rss content order by content nums
desc ,id desc limit 0,30";
$result = mysql query ( $sql, $mysql );
$nums = mysql num rows ( $result );
//for 循环列出所有的 rss 源
$rss content list = "";
for($i = 0; $i < $nums; $i ++) {
    $rss content [$i] = mysql fetch array ( $result );
    $rss content list .= "<tr><td>" . $rss content [$i] ["id"] . "</td><td>" .
    $rss content [$i] ["title"] . "</td><td>" . $rss content [$i] ["source"] .
    "</td><td>" . "<a href=\"del_rss_content.php?id=" . $rss_content [$i]
    ["id"] . "\"onClick='return confirm(\" 你真的要删除该文章么？\");'>删除
    </a></td></tr>";
}
$smarty->assign ( "rss content list", $rss content list );    //列表标签
$smarty->display ( "./rss_content_list.tpl" ); //取指定条目的 rss 源列表结束
mysql close ( $mysql );
?>
```

【代码解读】

由于上面的代码实现的功能复用程度不高，所以直接执行 SQL 语句将指定条目的“RSS 源信息”列表取出。

RSS 头条管理程序对应的 HTML 页面模板代码（rss_content_list.tpl）的设计如下：

```
<table width="100%" border="0" cellpadding="0" cellspacing="0"
    id="mastPath">
```

```
    <tr>
        <td>您的位置：<a href="loginindex.php">首页</a> &raquo; <a
            href="article_list.php">娱乐联播</a> &raquo; RSS 源列表管理</td>
    </tr>
</table>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td valign="top" class="Left">
        <table width="100%" border="0" cellspacing="0" cellpadding="0">
            <tr>
                <td>&oplus; <span class="Boo"><a href="rss catalog. php">RSS
                栏目管理</a></span>
                </td>
            </tr>
            <tr>
                <td>&oplus; <a href="new rss source.php">添加 RSS 源</a></td>
            </tr>
            <tr>
                <td>&oplus; <a href="rss_source_list.php">已有 RSS 源管理</a>
                </td>
            </tr>
            <tr>
                <td>&oplus; <a href="rss content list.php">RSS 头条管理
                </a></td>
            </tr>
        </table>
        </td>
        <td class="main">
        <form name="form2" method="get" action="rss_source_list.php">
        <table width="100%" border="0" align="center" cellpadding="0"
            cellspacing="0" class="mainHead">
            <tr>
                <td width="30%">// RSS 头条管理 //</td>
            </tr>
        </table>
        </form>
        <table width="100%" border="0" cellpadding="0" cellspacing="1"
            class="TableCenter">
            <tr class="Tit">
                <td width="12%">RSS 文章编号</td>
                <td width="60%">RSS 文章标题</td>
                <td width="13%">RSS 源</td>
                <td width="15%">操作</td>
            </tr>
            {|$rss content list|}
        </table>
        <table width="100%" height="30" border="0" cellpadding="0"
            cellspacing="0">
            <tr>
                <td width="50%"></td>
                <td width="50%" align="right"></td>
            </tr>
        </table>
        </td>
    </tr>
</table>
```

在左侧控制面板中选择“RSS 头条管理”，或者在浏览器中执行下面的链接：

```
http://localhost/lb/admin/rss_content_list.php
```

RSS 头条管理界面程序的浏览效果，如图 18.8 所示。



图 18.8　RSS 头条管理

（2）创建删除指定文章的 PHP 脚本文件，命名为 del_rss_content.php。代码如下：

```
<?php
//详细介绍：删除指定的 rss 文章
include_once (dirname ( __FILE__ ) . "/include.smarty.php");//取得 smarty 类
include_once (dirname ( __FILE__ ) . "/include.mysqlconfig.php");
                                                //取得 mysql 参数
include_once (dirname ( __FILE__ ) . "/include.rssfuns.php");
                                                //取得 RSS 的函数库
//创建数据库连接
$mysql = mysql_connect ( $mysqlconf ["host"], $mysqlconf ["user"],
$mysqlconf ["pass"] );
mysql_select_db ( $mysqlconf ["cmsdb"] );
//获得要删除的文章 id
$id = var_process ( "get", "id" );
if ($id == "") {
    header ( "Location: rss_content_list.php" );   //浏览器定向到 RSS 源列表
    exit ();
}
else {
    $sql = "delete from rss_content where id='" . $id . "'";
    $result = mysql_query ( $sql, $mysql );
    if ($result)                                //判断数据库状态变量返回结果
        echo "RSS 文章删除成功";
        echo "<a href=\"rss_content_list.php\">查看源列表</a>";
        exit ();
    } else {                                    //返回错误抛出错误提示
        echo "RSS 文章删除失败";
        echo "<a href=\"rss_content_list.php\">返回重试</a>";
        exit ();
    }
}
mysql_close ( $mysql );                         //关闭数据库连接
?>
```

选择指定的文章点击“删除”链接，程序输入如下：

```
RSS 文章删除成功
```

18.4.5 RSS 信息源抓取

RSS 信息抓取部分的主要内容是介绍 RSS 信息抓取的原理。通过对 RSS 信息抓取的原理讲解，结合在前文介绍过的准备知识，完成娱乐联播模块中的 RSS 抓取程序。这里读者需要着重理解 RSS 的抓取原理，以及在具体的业务模型中如何使用 PHP 实现 RSS 信息抓取。

1. RSS信息源抓取原理

通常网站管理者会在网站上显示 RSS 内容，或提供相关内容的其他网站的标题和摘录，这样便于访问者可以在集中的指定频道看到许多网站的头条新闻，这就是所谓的内容聚合最直观的体现。

需要注意的是，并不是将抓取的 RSS 内容直接转换成 HTML 页面代码，具体的步骤是：

（1）使用 RSS 的处理类获取目标源地址的内容。

（2）在处理数据信息并将内容部分按照指定的样式展示（输出到浏览器）。

在本章中，将使用一套开放源代码 PHP 类，称为 Magpie（http://magpierss.sourceforge.net/），从 RSS 文档中来提取想要的内容。

下面是一个标准的 RSS 目标源 feed 文件的格式，这就是 RSS 类需要解析的 feed 文件的格式样式。

```
<?xml version="1.0" encoding="UTF-8"?>
    <rss version="2.0" xmlns:dc=http://purl.org/dc/elements/1.1/">
    <channel>
    <title>Crazy Monkey News</title>
  <link>http://www.crazymonkeynews.com</link>
  <language>en-us</language>
  <description>The craziest news about monkeys</description>
    <item>
        <title>Title</title>
        <description>description </description>
        <pubDate>Fri, 10 Aug 2007</pubDate>
    <author>author</author>
    <guid isPermaLink="true">http://www.crazymonkeynews.com/ bluemonkeys
    </guid>
    <link>http://www.crazymonkeynews.com/bluemonkeys</link>
    <category>Monkey News</category>
    </item>
  </channel>
</rss>
```

【代码解读】

从上面的代码可以看出，RSS 的标签是使用标准的 XML 标签格式。标签中的内容包括文章标题、编码语言、作者和内容描述等常规内容信息。

解析和显示 RSS feed 由如下的 3 个部分组成：

（1）获得 Magpie RSS 类，在 http://magpierss.sourceforge.net/下载。

（2）使用 Magpie 类将获得 RSS 源中的内容。

（3）使用 PHP 处理获得信息（写入数据库等）。

程序的执行基本流程如下：

（1）加载文件名为 parseRSS.php 的 PHP 文件，初始化 Magpie RSS 解析函数。这个函数通过 Magpie 类中的 fetch_rss()函数获得 RSS 源文件中的内容信息。

（2）创建文件名为 read_rss.php 的 PHP 脚本文件，通过 fetch_rss()函数来获得 RSS 源中的内容，并将内容整理、过滤后写入数据库。

（3）当 fetch_rss()函数被调用时，会以一系列属性的对象（成员变量）的形式返回 feed。每个属性代表了 RSS feed 中的一个标签，将和原来的标签包含相同的文字。

注意：调用 fetch_rss()函数，当抓取 feed 有问题的时候，需要使用一个简单的判断条件来制止进程，避免程序异常。

2．RSS抓取程序

RSS 抓取程序是娱乐联播模块的核心程序，所有的内容都是由这个程序产生。该程序的核心是对获取内容的处理和对异常情况的流程控制，读者在学习时注意如何获取内容信息和使用控制语句处理异常情况。创建一个用来获得 RSS 源信息，并将结果处理后写入数据库的 PHP 脚本文件（即 RSS 抓取程序），命名为 read_rss.php。该文件代码如下：

```
<?php
//详细介绍：读取 RSS 源的程序并将更新后的每个 RSS 源所读取出的新记录为首页显示
include_once (dirname ( __FILE__ ) . "/include.mysqlconfig.php");
                                               //取得 mysql 参数
include_once (dirname ( __FILE__ ) . "/include.rssfuns.php");
                                               //取得检测变量有效性的函数
require_once (dirname ( __FILE__ ) . "/rss_fetch.inc");
define ( 'MAGPIE_OUTPUT_ENCODING', 'UTF-8' );
define ( 'MAGPIE_DEBUG', 1 );
define ( 'MAGPIE_CACHE_AGE', 10 );
//创建数据库链接
$mysql = mysql_connect ( $mysqlconf ["host"], $mysqlconf ["user"], $mysqlconf
["pass"] );
mysql_select_db ( $mysqlconf ["cmsdb"] );
$cataid = "1";
$cataid1 = get_ids ( $cataid, $mysql );          //取得所有分类 id 以,号连接
$son_cataid = str_replace ( $cataid . ",", "", $cataid1 );//将本身的 id 去掉
$rss_source_list = sel_source ( $son_cataid, $mysql );//找出所有的 rss_source
$rss_source_nums = count ( $rss_source_list ); //计算 RSS 源数量
//遍历 RSS 信息源数据集
for($i = 0; $i < $rss_source_nums; $i ++) {
    $now_time = time ();    //取得当前时间，时间格式为 UNIX 时间戳格式
    $update_time = strtotime ( $rss_source_list [$i] ["update_time"] );
                                   //取得记录上次抓取时间，将时间格式转化为 unix 时间戳格式
    $get_time = $rss_source_list [$i] ["get_time"];//取得抓取间隔时间
    if ($now_time - $update_time >= $get_time)
                                   //如果当前时间与上次抓取时间超过间隔时间，则抓取
{
        if ($rss_source_list [$i] ["rss_url"] != "") {
```

```
            //读取 RSS 源内容
            $rss info = fetch rss ( $rss source list [$i] ["rss url"] );
            if ($rss_info) //如果读到数据，则进行下一步
{
                //以下为更新数据库中 RSS 读取的时间
                $sql = "update rss source set update time=now() where id='" .
                $rss_source_list [$i] ["id"] . "'";
                $result = mysql query ( $sql, $mysql );
                //以下为将读取到的记录写入数据库
                foreach ( $rss info->items as $rss content )
                        //将读取到的数据提取到一维数组里并写入
{
                    //转换 title 编码格式
                    $rss content ['title'] = @iconv ( "UTF-8", "GB2312",
                    $rss content ['title'] );
                    //字符转义，避免 SQL 注入攻击
                    $rss content ['title'] = str replace ( "'", "\'", $rss
                    content ['title'] );
                    $rss content ['title'] = str replace ( ",", "\,", $rss
                    content ['title'] );
                    $rss_content ['title'] = preg_replace ( "/<[\/]*?[^<>]
                    *?>/", "", $rss_content ['title'] );
                    //如果 description 值不存在或为空，则默认为空
                    $rss content ['description'] = isset ( $rss content
                    ['description'] ) ? $rss_content ['description'] : "";
                    $rss content ['description'] = @iconv ( "UTF-8","GB2312",
                    $rss_content ['description'] );//转换 description 编码格式
                    $rss content ['description'] = str replace ( "'", "\'",
                    $rss_content ['description'] ); //替换危险字符
                    $rss content ['description'] = str replace ( ",", "\,",
                    $rss content ['description'] );
                    //查询读取的记录在同一个分类下是否有记录，（以读取到的 URL 为查找
                     对象）
                    $sql = "select id from rss_content where catalogid ='" .
                    $rss source list [$i] ["catalogid"] . "' and url='" .
                    $rss_content ["link"] . "'";
                    $result = mysql query ( $sql, $mysql );
                    $check rss source = mysql fetch array ( $result );
                    //如果没有则应该无记录存在，而且读取到的 RSS 标题不为空，则写入
                     数据库
                    if ($check_rss_source ["id"] == "" and $rss_content
                    ["title"] != "") {
                        //将读取到的数据写入数据库
                        $sql = "insert into rss content (catalogid,title,
                        url,description,source id,source,content time,
                        content_status)values" . "('" . $rss_source_list
                        [$i] ["catalogid"] . "','" . $rss content ["title"] .
                        "','" . $rss_content ["link"] . "'," . "'" . $rss_
                        content ['description'] . "','" . $rss source list
                        [$i] ["id"] . "','" . $rss source list [$i] ["source"].
                        "',now(),'1')";
                        $results = mysql query ( $sql, $mysql );
                    }
                }
            }
        }
    }
}
```

【代码解读】

以上代码完成了读取 RSS 源的程序，并将结果整理到数组中，读者需要注意读取过程中条件的判断。下面介绍将更新后的每个 RSS 源所读取出的新记录为首页显示部分，具体的代码如下：

```
//以下为将首页显示的 RSS 记录替换为最新的记录程序
$cataid = "1";
$cataid1 = get_ids ( $cataid, $mysql );                    //取得所有子分类
$son_cataid = str_replace ( $cataid . ",", "", $cataid1 );
                                                //将本身的 id 去掉，父分类没有值
$rss_source_list = sel_source ( $son_cataid, $mysql );//获得所有的 rss_source
$rss_source_nums = count ( $rss_source_list );         //计算有多少个 RSS 源
for($i = 0; $i < $rss_source_nums; $i ++) {
    //查询所有最新的 rss_content 的记录，以 RSS 源记录排序
    $sql = "select id from rss_content where source_id=" . $rss_source_list
    [$i] ["id"] . "  order by id desc limit 1";
    $result = mysql_query ( $sql, $mysql );
    //取出该 RSS 源最新更新记录的第一条
    $content = mysql_fetch_array ( $result );
    //如果该 RSS 源有记录，则进行下一步，否则不执行，如果有记录但是没更新.
    if ($content ['id'] != "") {
        //修改这条最新的记录 设置其为首页显示状态 set index_status='1'
        $sql = "update rss_content set index_status ='1' where id='" .
        $content ["id"] . "'";
        $result = mysql_query ( $sql, $mysql );
        //以下为连接各 RSS 源设置为显示状态的 id，第一个记录前不能加逗号连接，所以判断
          了一下
        if ($id == "") {
            $id .= $content ['id'];
        } else {
            $id .= "," . $content ['id'];
        }
    }
}
//将首页显示为 1 的状态修改为 0，只需要排除最新更新的就可以了
$sql = "update rss_content set index_status='0' where id not in(" . $id . ") ";
$result = mysql_query ( $sql, $mysql );
mysql_close ( $mysql );
?>
```

【代码解读】

程序首先获取数据库中的 RSS 源地址，然后通过前面引入的 fetch_rss()函数，获取对应的内容。代码实现如下：

```
//读取 RSS 源内容
$rss_info = fetch_rss( $rss_source_list[$i]["rss_url"] );
```

然后使用 str_replace()函数对内容中的特殊字符做转换，避免写入数据库的时候产生错误。代码实现如下：

```
//转换 title 编码格式
$rss_content['title']=@iconv("UTF-8","GB2312",$rss_content['title']);
//字符转义，避免 SQL 注入攻击
$rss_content['title']=str_replace("'","\'",$rss_content['title']);
$rss_content['title']=str_replace(",","\,",$rss_content['title']);
```

```
$rss_content['title']=preg_replace("/<[\/]*?[^<>]*?>/","",$rss_content['title']);
//如果 description 值不存在或为空，则默认为空
$rss_content['description']=isset($rss_content['description'])? $rss_
content['description']: "" ;
//转换 description 编码格式
$rss_content['description']=@iconv("UTF-8","GB2312",$rss_content['descr-
iption']);
$rss_content['description']=str_replace("'","\'",$rss_content['descript-
ion']);
$rss_content['description']=str_replace(",","\,",$rss_content['descripti-
on']);
```

18.5　娱乐联播频道

娱乐联播频道是按不同的频道主题汇聚网络中的资讯，提供用户访问浏览的频道栏目。该频道是典型的信息聚合的应用模式。在娱乐联播的首页汇聚了全部频道栏目下的文章信息，这些信息是通过 RSS 抓取程序从目标源网站获得的，娱乐联播的栏目页是显示对应栏目的具体的资讯的。

18.5.1　娱乐联播主首页

娱乐联播首页程序负责抓取文章内容，并按照栏目分类显示，该程序也是模块的入口程序。由于娱乐联播首页时效性很强，而且根据需求经常要变动首页的显示方式，所以在程序设计上采用加载配置文件的方式来实现。这样最大程度上保证了程序的可扩展性和排版的灵活性。

创建文件名为 include.config.php 的配置文件，配置文件的内容样式如下：

```
//八卦 列表的参数
$fun_list[0]["tag_name"]="bagua_list"; //标签名
$fun_list[0]["catalog"]=2;             //所取分类 id
$fun_list[0]["offset"]=0;              //从第几个取
$fun_list[0]["num"]=5;                 //取多少条
$fun_list[0]["title_length"]=23;       //标题字长度
//音乐 列表的参数
$fun_list[1]["tag_name"]="music_list"; //标签名
$fun_list[1]["catalog"]=3;             //所取分类 id
$fun_list[1]["offset"]=0;              //从第几个取
$fun_list[1]["num"]=4;                 //取多少条
$fun_list[1]["title_length"]=23;       //标题字长度
```

【代码解读】

娱乐联播程序通过读取配置文件中参数信息决定显示栏目、显示条数、标题字长度和开始编号等基本信息。下面介绍娱乐联播首页程序执行过程，程序代码如下：

```
<?php
//详细介绍：娱乐联播首页
include_once (dirname ( __FILE__ ) . "/include.smarty.php");//取得 smarty 类
include_once (dirname ( __FILE__ ) . "/include.mysqlconfig.php");
```

```
                                          //取得 mysql 参数
include once( dirname(  FILE  )."/include.rssfuns.php" );//取得 RSS 的函数库
include_once( dirname(__FILE__)."/include.config.php" );
                                          //取得 config 配置文件
mb_internal_encoding("gb2312");//设定切分字符串函数的文字编码
//创建数据库链接
$mysql = mysql connect( $mysqlconf[ "host" ], $mysqlconf[ "user" ],
$mysqlconf[ "pass" ] );
mysql_select_db( $mysqlconf[ "cmsdb" ] );
//连接检查
if(!$mysql)
{
    exit ;
}
```

【代码解读】

上述代码加载了程序所需要的函数库文件，然后设定切分字符串函数的文字编码，初始化数据库连接。下面介绍调用配置文件生成文章内容的代码实现，读者在这部分注意梳理判断逻辑的嵌套，具体的代码如下：

```
//分别处理 config 配置中的数据
for($i=0;$i< count($top list);$i++)
{
    $list[$i]="";
    $config=$top_list[$i]; //将 config 中二维数组配置记录转化为 1 维数组方便操作
    $son cataid=$config["catalog"];    //取得查询分类的 id
    //查询数据库里的对应分类的所有记录
    $content=top content($son cataid,$config["offset"],$config["num"],$mysql);

    $content nums=count($content);     //取得记录数目
    for($j=0;$j< $content_nums;$j++)
    {
        $content[$j]["title"]=mb substr($content[$j]["title"],0,$config
        ["tit   le_length"]);
        $list[$i].="<li>•<a href=\"content-".$content[$j]["id"].".html\"
        target=\"_blank\">".$content[$j]["title"]."</a></li>" ;
    }
    //如果记录数不够 config 中所需要的条数
    if($content nums< $config["num"])
    {
        $fill_nums=$config["num"]-$content_nums ;  //计算差多少条
        $content2=content list($son cataid,1,$fill nums,$mysql);
                    //补足数目，以该分类下的第二条最新的记录开始补
        $content nums2=count($content2);
        for($k=0 ;$k< $content_nums2;$k++)
        {
        $content[$i]["title"]=mb substr($content[$i]["title"],0,17);
                                          //截取字数，方便显示
        $top list.="<li>•<a href=\"content-".$content[$i]["id"].".html\"
        target=\" blank\">".$content[$i]["title"]."</a></li>" ;
        }
    }
}
```

【代码解读】

通过配置文件的内容可以了解到，配置文件就是将参数生成一个标签数组。因此要计

算每个标签中的个数，然后使用 for()循环函数取出对应的数据。代码如下：

```
for($i=0;$i< count($top list);$i++)
{
    $smarty->assign($top list[$i]["tag name"],$list[$i]);
                        //生成该 config 配置文件中的 smarty 标签的内容
}
unset($list);
for($i=0;$i< count($all list);$i++)        //分别处理 config 配置中的数据
{
    $list[$i]="";
    $config=$all_list[$i]; //将 config 中二维数组配置记录转化为一维数组方便操作
    $cataid=$config["catalog"];
    $son_cataid=get_ids($cataid,$mysql);   //取得查询分类的 id
    $son cataid=str replace($cataid.",","",$son cataid);//取得该分类下面的子分类
    //将父节点去掉，如果没有子类，则不会替换
    $content=index list($son cataid,$config["offset"],$config["num"],$my sql);
                                        //查询数据库里的对应分类的所有记录
    $content_nums=count($content);           //取得记录数目
    for($j=0;$j< $content nums;$j++)        //生成 smarty 标签内容
    {
        $content[$j]["title"]=mb substr($content[$j]["title"],0,$config
        ["title length"]);
        $list[$i].="<li>•<a href=\"content-".$content[$j]["id"].".html\"
        target=\" blank\">".$content[$j]["title"]."</a></li>" ;
    }
    if($content_nums< $config["num"])  //如果记录数不够 config 中所需要的条数
    {
        $fill_nums=$config["num"]-$content_nums ;  //计算差多少条

        $content2=content list($son cataid,1,$fill nums,$mysql);
                            //补足数目，以该分类下的第二条最新的记录开始补
        $content nums2=count($content2);
        for($k=0 ;$k< $content nums2;$k++)
        {
                $content2[$k]["title"]=mb substr($content2[$k]["title"],0,
                $config["title_length"]);       //截取标题字数，方便显示
                $list[$i].="<li>•<a href=\"content-".$content[$k]["id"].
                ".html\"target=\" blank\">".$content[$k]["title"]."</a>
                </li>" ;
        }
    }
}
for($i=0;$i< count($all list);$i++)
{
    //生成该 config 配置文件中的 smarty 标签的内容
    $smarty->assign($all_list[$i]["tag_name"],$list[$i]);
}
//取消$list 变量，防止数组越界
unset($list);
```

【代码解读】

以上代码获得对应栏目的内容后，将内容数组生成对应名字的 Smarty 标签，这里也是通过遍历的方式循环生成标签。代码样式如下所示：

```
for($i=0;$i< count($all_list);$i++)
{
```

```
    //生成该 config 配置文件中的 smarty 标签的内容
    $smarty->assign($all list[$i]["tag name"],$list[$i]);
}
//取消$list 变量，防止数组越界
unset($list);
```

需要注意的是，要使用 unset()函数对$list 变量注销，防止数组越界影响在下面程序中的使用。下面继续分别处理每块新闻栏目，这部分处理的过程比较繁复，读者需注意理解使用 for 循环和 if 语句控制业务流程。相关的代码如下：

```
//分别处理每块新闻栏目表
for($i=0;$i< count($fun list);$i++)
{
    $list[$i]="";
    $config=$fun_list[$i];          //将新闻的所有参数（一维数组），给 new-list
    $cataid=$config["catalog"];     //设定分类的 id
    $son cataid=get ids($cataid,$mysql);   //取得该分类下面的子分类
//  $son cataid=str replace($cataid.",","",$son cataid);
                                    //将父节点去掉（因为没有文章）
$content=index list($son cataid,$config["offset"],$config["num"],$mysql);
                                    //查询数据库里的对应分类的所有文章
    $content nums=count($content); //取得记录数目
    //循环连接成网页上显示的文字
    for($j=0;$j< $content nums;$j++)
    {
$content[$j]["title"]=mb substr($content[$j]["title"],0,$config["title length"]);
        $content[$j]["content_time"]=mb_substr($content[$j]["content_time"],5,11);
        $content[$j]["source"]=mb substr($content[$j]["source"],0,8);
        $list[$i].="<tr><td>•<a href=\"content-".$content[$j]["id"].
        ".html\" target=\" blank\">".$content[$j]["title"]."</a>"
                        ."<span class=\"posttime\">  [".$cont-
                        ent[$j]["content time"] ."]</span></td>"
                        ."<td class=\"sourceName\">".$content[$j]["source"].
                        "</td></tr>";
    }
    if($content nums< $config["num"])
                //如果取得的记录条数少于 config 中需求条数则继续取
    {
        $fill nums=$config["num"]-$content nums ;  //取的数量是缺少的数量
        $content2=content list($son cataid,1,$fill nums,$mysql);
        $content_nums2=count($content2);           //计算条数
        for($k=0 ;$k< $content nums2;$k++)
        {
        $content2[$k]["title"]=mb_substr($content2[$k]["title"],0, $config
        ["title length"]);
        $content2[$k]["content time"]=substr($content2[$k]["content  time"],5,11);
        $content2[$k]["source"]=mb substr($content2[$k]["source"],0,8);
        $list[$i].="<tr><td>•<a href=\"content-".$content2[$k]["id"]."
        .html\" target=\" blank\">".$content2[$k]["title"]."</a>"
                                ."<span class=\"posttime\">  [".$co-
                                ntent2[$k]["content time"]."]</span></td>"
                                ."<td class=\"sourceName\">".$content2[$k]
```

```
                                ["source"]."</td></tr>";
        }
    }

}
//循环生成模板列表标签
for($i=0;$i< count($fun_list);$i++)
{
    //生成该栏目文章列表的 smarty 标签的内容
    $smarty->assign($fun_list[$i]["tag_name"],$list[$i]);
}
//加载首页模板
$smarty->display("index.tpl");
//关闭数据库链接
mysql_close($mysql);
?>
```

上面的程序已经获得了首页的频道数据，并整理成指定格式的 Smarty 标签。下面介绍 HTML 模板页面模板（index.tpl）的设计，众多的频道区块在代码方面差别很小，这里以其中一个部分为例进行描述，代码如下：

```
<div class="main">
    <div class="rssDiv">
        <div class="rssLeft">
            <div class="rssTitle">
                <div class="rssTitle_1"><a href="list.php?catalog=2"> .八
                卦._</a></div>
                <div class="rssTitle_2"><a href="list.php?catalog=2">更多
                ...</a></div>
            </div>
            <div class="rssBox">
                <table width="100%" border="0" cellspacing="0"cellpadding="0">
                {|$bagua_list|}
              </table>
            </div>
        </div>
    </div>
    <div class="rssDiv2">
        <div class="rssRight">
            <div class="rssTitle">
                <div class="rssTitle_1"><a href="list.php?catalog=4"> .音
                乐._</a></div>
                <div class="rssTitle_2"><a href="list.php?catalog=4">更多
                ...</a></div>
            </div>
            <div class="rssBox">
               <table width="100%" border="0" cellspacing="0"cellpadding="0">
               {|$music_list|}
```

```
            </table>
           </div>
        </div>
    </div>
</div>
```

在浏览器中执行下面的链接：

```
http://localhost/lb/
```

娱乐联播首页界面程序的浏览效果如图 18.9 所示。

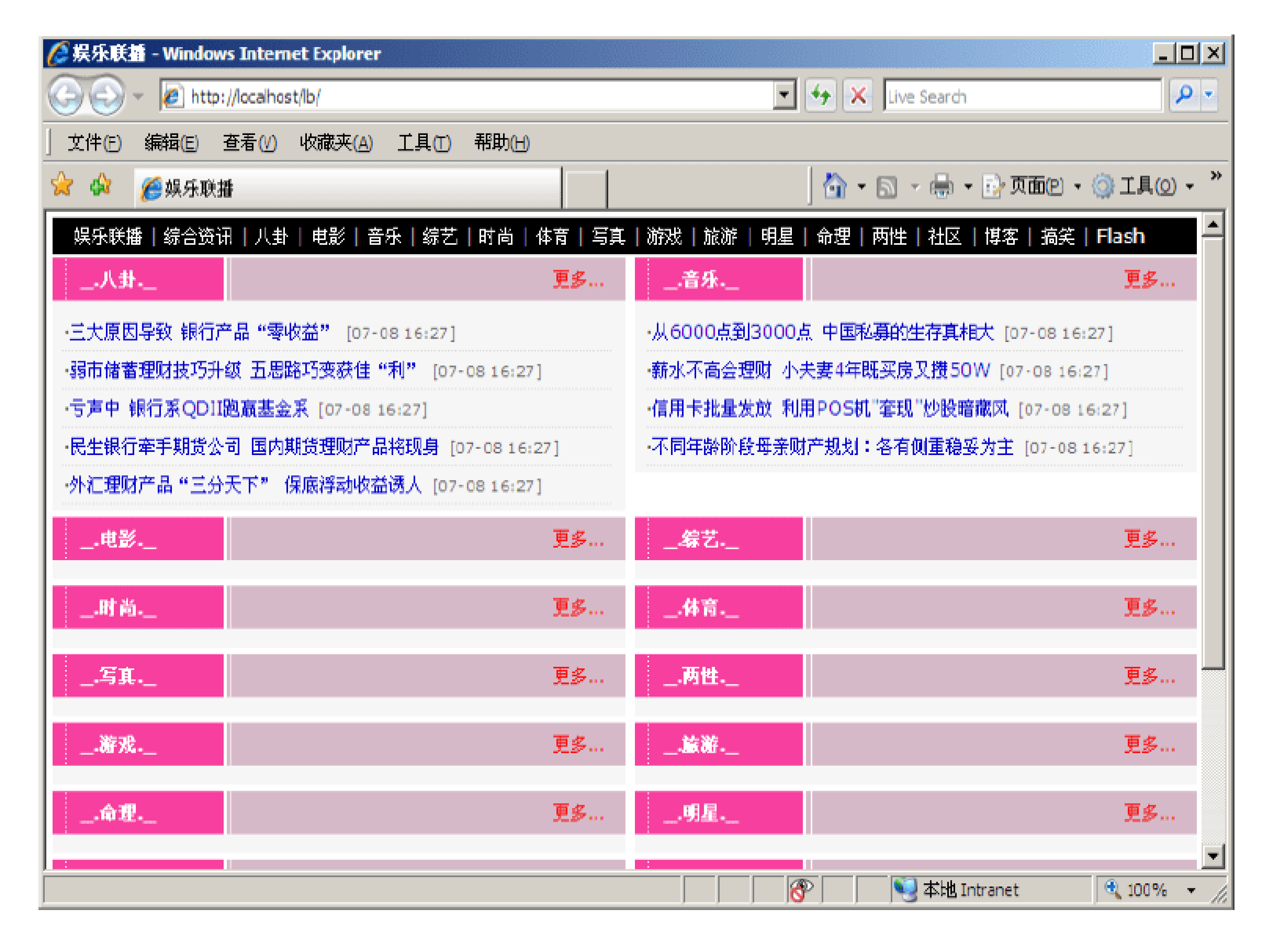


图 18.9　娱乐联播首页

18.5.2　娱乐联播栏目

娱乐联播列表页面程序与首页的展示逻辑有所不同，栏目的内容是固定的，不用像首页程序那样需要灵活处理和修改，所以在程序设计上有所不同，主要实现对指定栏目内容的遍历和页面展示。下面介绍具体的代码实现：

```
<?php
//详细介绍：RSS 抓取内容列表页
mb_internal_encoding ( "gb2312" ); //提供给切分字符串函数的文字编码
include_once (dirname ( __FILE__ ) ."/include.smarty.php");//取得 smarty 类
include_once (dirname ( __FILE__ ) ."/include.mysqlconfig.php");
                                    //取得 mysql 参数
include_once (dirname ( __FILE__ ) . "/include.rssfuns.php");
                                    //取得 RSS 的函数库
// 创建数据库链接
$mysql = mysql_connect ( $mysqlconf ["host"], $mysqlconf ["user"],
```

```
$mysqlconf ["pass"] );
mysql select db ( $mysqlconf ["cmsdb"] );
$catalog = var process ( "get", "catalog", 1 );
            //取得下拉菜单的被选中项 id，如果没有，则默认为娱乐联播
$cata tree = getPathIdArray ( $catalog, $mysql );
            //获得栏目名 title 导航菜单链接
$cata tree num = count ( $cata tree );
$list title = $cata tree [$cata tree num - 1] ["cataname"];
$smarty->assign ( "list title", $list title );
$cata_list = catamenu ( $cata_tree );                  //获得分类
$cata_list .= "文章列表";
$smarty->assign ( "second menu", $cata list );
//列表分类部分
//取得下拉菜单中被选中项的子 ID
$son cataid = get ids ( $catalog, $mysql );
$content_nums = content_nums ( $son_cataid, $mysql );  //获得分类文章总数
$currpage = var process ( "get", "page", 1 );          //获得参数当前页数
$itemnum = 20;                                         //单页条目数
$listpage = divpage_toadmin ( $content_nums, $currpage, $itemnum );//获得页码
$list offset = $listpage ["rec start"];                //偏移值
$firstpage = "<a href=\"list.php?catalog=" . $catalog . "&&page=" . $listpage
["firstpage"] . "\">首 页</a>";
$prepage = "<a href=\"list.php?catalog=" . $catalog . "&&page=" . $listpage
["prepage"] . "\">上一页</a>";
$nextpage = "<a href=\"list.php?catalog=" . $catalog . "&&page=" . $listpage
["nextpage"] . "\">下一页</a>";
$lastpage = "<a href=\"list.php?catalog=" . $catalog . "&&page=" . $listpage
["max_page"] . "\">尾 页</a>";
$smarty->assign ( "content_nums", $content_nums );        //总条数标签
$smarty->assign ( "page nums", $listpage ["max page"] );  //生成总页数标签
$smarty->assign ( "first_page", $firstpage );             //首页标签
$smarty->assign ( "last page", $lastpage );               //尾页标签
$smarty->assign ( "pre_page", $prepage );                 //上一页标签
$smarty->assign ( "next_page", $nextpage );               //下一页标签
$smarty->assign ( "now_page", $currpage );                //当前页标签
```

【代码解读】

以上代码获得栏目名、“title 导航菜单链接”和分页相关参数等。在准备好这部分信息后，可以继续取得该频道下面 RSS 的列表。代码在实现上采用“分块”的模式是为了便于程序具有更好的“可读性”，具体的代码如下：

```
//取指定条目的 rss 源列表
//取得该子 id 下的所有 rss 源
$content = content list ( $son cataid, $list offset, $itemnum, $mysql );
$content_nums = count ( $content );         //计算总 rss 源共有多少条
$content_list = "";
```

```
//for 循环列出所有的 rss 源
for($i = 0; $i < $content nums; $i ++) {
    $content list .= "<tr>" . "<td>•<a href=\"content-" . $content [$i]
    ["id"] . ".html\" target=\" blank\">" . $content [$i] ["title"] . "</a>
    " . "<span class=\"posttime\">[" . $content [$i] ["content time"] .
    "]</span></td>" . "<td class=\"sourceName\">" . $content [$i]
    ["source"] . "</td></tr>";
}
$smarty->assign ( "content list", $content list );//列表标签
$cataid = "1";
$son cataid = get ids ( $cataid, $mysql );          //取得该分类下面的子分类
$content = content list ( $son cataid, 0, 10, $mysql );
                                       //查询数据库里的对应分类的所有文章
$content_nums = count ( $content );    //取得列数大小
$top_list = "";
$top source = "";
//循环连接成网页上显示的文字
for($i = 0; $i < $content nums; $i ++) {
    $content [$i] ["title"] = mb substr ( $content [$i] ["title"], 0, 16 ) .
    "...";  //取标题前 28 个汉字
    $top list .= "<li>•<a href=\"content-" . $content [$i] ["id"] . ".html\"
    target=\"_blank\">" . $content [$i] ["title"] . "</a></li>";
}
$smarty->assign ( "top list", $top list );
//综合资讯开始
$cataid = "1";                                  //设定分类的 id
$son_cataid = get_ids ( $cataid, $mysql );      //取得该分类下面的子分类
//$son cataid=str replace($cataid.",","",$son cataid);
                                       //将父节点去掉（因为没有文章）
$content = index_list ( $son_cataid, 0, 15, $mysql );
                                       //查询数据库里的对应分类的所有文章
$content_nums = count ( $content );    //取得列数大小
$list = "";
for($i = 0; $i < $content nums; $i ++) //循环连接成网页上显示的文字
{
    $content [$i] ["title"] = mb substr ( $content [$i] ["title"], 0, 17 );
    $list .= "<li>•<a href=\"content-" . $content [$i] ["id"] . ".html\"
    target=\" blank\">" . $content [$i] ["title"] . "</a></li>";
}
if ($content nums < 15) {
    $fill nums = 15 - $content nums;
    $content2 = content list ( $son cataid, 1, $fill nums, $mysql );
    $content_nums2 = count ( $content2 );
    for($j = 0; $j < $content nums2; $j ++) {
        $content2 [$j] ["title"] = mb substr ( $content2 [$j] ["title"], 0, 17 );
        $list .= "<li>•<a href=\"content-" . $content [$j] ["id"] . ".html\"
        target=\" blank\">" . $content [$j] ["title"] . "</a></li>";
    }
}
$smarty->assign ( "all list", $list );//生成该栏目文章列表的 smarty 标签的内容
$smarty->display ( "list.tpl" );       //加载页面模板
mysql_close ( $mysql );                //关闭数据库链接
?>
```

【代码解读】

在上面的代码中，程序一共完成了 3 个方面的工作。

（1）首先将对应栏目的数据取出，代码如下：

```
//取指定条目的 rss 源列表
//取得该子 id 下的所有 rss 源
$content = content_list( $son_cataid,$list_offset,$itemnum,$mysql);
```

（2）将所有取出的信息用 for 循环整理成对应的列表标签。

（3）将内容信息输出到页面模板展示。下面介绍程序对应 HTML 模板页面模板（list.tpl）部分的设计核心代码如下：

```
<!--加载 header-->
{|include file="head.tpl"|}
<!--加载栏目导航-->
{|include file="second_menu.tpl"|}
<div class="main">
    <div id="leftDiv">
        <div class="face">
            <h3>24 小时内最热文章</h3>
            <ul>{|$top_list|}</ul>
        </div>
        <div class="face">
            <h3>综合资讯</h3>
            <ul>{|$all_list|}</ul>
        </div>
    </div>
<div id="mainDiv">
    <div class="rssTitle">
    <div class="rssTitle_1">
        <a href="#">_.列表._</a>
    </div>
    <div class="rssTitle_2">
        <a href="javascript:;"> </a>
    </div>
</div>
<div class="rssBox">
    <table width="100%" border="0" cellspacing="0" cellpadding="0">
        <!--加载资讯 list-->
        {|$content_list|}
     </table>
</div>
<!--分页参数设置开始-->
<div id="rssPage">共 {|$content_nums|} 篇     
{|$first_page|} {|$pre_page|} {|$next_page|} {|$last_page|}  
  20/页  共{|$page_nums|}页 当前第{|$now_page|}页</div>
<!--分页参数设置结束-->
    </div>
</div>
```

在娱乐联播首页单击“更多”链接，或者在浏览器中执行下面的链接：

```
http://localhost/lb/list.php?catalog=2
```

娱乐联播首页界面程序的浏览效果，如图 18.10 所示。



图 18.10 娱乐联播栏目列表

18.6 小 结

娱乐联播模块是对 RSS 内容聚合的一个具体的应用，在技术上的要点是使用 PHP 实现对 RSS 源的信息抓取。本章使用 Magpie 类（开源的 RSS 解析类）解析并获得 RSS 源中的内容。

本章首先介绍娱乐联播模块的基础设计，然后介绍了 RSS 基本概念，接着对娱乐联播流程做了具体的描述。第 18.2 节介绍了娱乐联播的数据库部分的设计。第 18.3 节介绍了娱乐联播模块应用到的核心函数，为后面程序实现具体的业务流程奠定了技术基础。第 18.4 节具体地介绍了娱乐联播流程后台程序的编码实现和界面设计，分别是 RSS 栏目管理、添加 RSS 源、已有 RSS 源管理、RSS 头条管理、RSS 信息源抓取。第 18.5 节通过娱乐联播频道介绍了抓取信息的展示过程和编码实现。

第 19 章　藏宝游戏（PHP+Flash+API）

藏宝游戏模块是基于 MySQL 数据库驱动的，PHP 与 Flash 文件交互的游戏应用程序。本章的重点内容是介绍 PHP 与 Flash 文件的交互原理和实现过程。读者通过本章的学习，可以扩展对 PHP 应用范围的认识，同时将会学习通过 API 接口编程来实现 PHP 与 Flash 的交互。

本章主要涉及的知识点如下。

- swf 文件：简单来说，就是使用 Flash 创作出的影片自己的特殊档案格式，swf 文件的整体结构是 header+body 的组成。
- session_destroy()：该函数的作用是销毁预先定义好的 Session 变量。
- API 封包：应用程序编程接口是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件的访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

19.1　藏宝游戏概述

藏宝游戏是一个 PHP 与 Flash 文件互动的页面游戏，这个游戏的设计目的是通过游戏的方式，增加与用户的互动，并在这个过程中产生付费等行为。本节将介绍藏宝游戏的玩法说明、游戏道具的作用描述和程序的工作流程。

19.1.1　藏宝游戏说明

在藏宝游戏中玩家已注册用户的身份登录系统进入游戏，用户可以在游戏地图区域的任意地方进行挖宝活动，并且有机会挖到道具和一些特殊的奖品。

如果用户挖到的是普通的道具，那么就可以直接使用或者直接放入用户的道具库。如果用户挖到的是奖品，要想得到这个奖品必须进行保存（可以在保存前进行移动），并且这个奖品在 10 分钟以内没有被其他用户挖到并且移动，用户在保存奖品的操作中不能够进行其他的操作。如果用户真的想得到所挖到的奖品，可以在保存过程中对道具进行加锁或者添加炸弹。

另外，初始用户有一定的“行动力”值和游戏币点数。用户每进行一次挖宝操作或者移动奖品操作都会消耗行动力值。游戏币点数主要是用来购买道具，如果用户用完了系统初始化的点数，用户可以通过积分兑换或者充值的方式重新补充体力值。

用户在第一次登录游戏的初始状态，系统会给他一个普通的挖宝道具（木铲子），挖宝道具是有一定的使用次数的（例如木铲子只能使用 30 次）。如果用户的挖宝道具报废了，需要购买新的挖宝道具。

19.1.2　游戏道具描述

在藏宝游戏中，对用户吸引最大的就是引入了游戏道具来增加游戏的趣味性，同时用户也更好地参与到游戏的互动中来。

具体的设定如下。

（1）用户首次登录的初始体力值：500，每次挖宝都会消耗一定的体力值，当没有体力时就不能继续挖宝。

（2）游戏币的兑换规则：100 游戏币=1 元人民币。

（3）用户每天登录系统会自动赠送 2 个单位的游戏币。

（4）当用户没有铲子时不能挖宝。

道具的作用和详细说明如表 19.1 所示。

表 19.1　游戏道具功能表

编号	道 具 名 称	功　　能	说　　明
1	木铲子	初始拥有	（10 游戏币）使用 30 次报废
2	钢铲子		（200 游戏币）使用 100 次报废
3	金铲子		（500 游戏币）不报废
4	小挂锁	可重复加锁。（开锁）	（2）加锁（2）（20 游戏币）
5	钢丝锁	可重复加锁。（开锁）	（6）加锁（6）（100 游戏币）
6	加密锁	可重复加锁。（开锁）	（10）加锁（10）（200 游戏币）
7	挂锁钥匙	可打开小挂锁，一把钥匙只能开一把锁	20 游戏币
8	钢丝钥匙	可打开钢丝锁，一把钥匙只能开一把锁	100 游戏币
9	加密钥匙	可打开加密锁，一把钥匙只能开一把锁	200 游戏币
10	万能钥匙	可打开任意锁	300 游戏币
11	米粉丝	+50 体力	10 游戏币
12	加蛋粉丝	+120 体力	20 游戏币
13	牛肉粉丝	+250 体力	40 游戏币
14	炸弹	–100 体力	50 游戏币
15	地雷	–200 体力	100 游戏币
16	TNT	–350 体力	200 游戏币
17	探宝器	使用时提示 3 个宝物的位置	100 游戏币
18	宝箱	装宝贝的箱子	

其中 1～18 为道具的编号，在 goods 表中对应着相应的 gid。

注意：里面对行动力的定义如果有所改动，只需要更改 goods 表中相应 nmovep 字段的值。

19.2　数据库设计

藏宝游戏的数据库的设计目的，是为了保存用户注册信息和游戏道具，以及获奖物品等。藏宝游戏模块的数据库中用到了 5 张表，分别是用户领奖信息表、道具及奖品信息表、物品详细信息表、用户资料信息表。

藏宝游戏的数据库表含义如下。

- 用户领奖信息表：记录用户领奖的信息。
- 道具及奖品信息表：记录游戏中道具、奖品情况的信息。
- 物品详细信息表：记录物品的详细信息。
- 用户资料信息表：记录用户现在所拥有的信息。
- 游戏用户的信息表：记录游戏用户的信息。

上面介绍了藏宝游戏的数据库各个表的具体含义，对每个表的功能作了详细的说明，下面来看具体的设计步骤。

首先建立 treasure 数据库，SQL 语句如下：

```
CREATE DATABASE ' treasure ';
```

19.2.1　用户领奖信息表

用户领奖信息表用来记录用户领奖的信息，在 treasure 数据库中建立一个命名为 drawing 的用户领奖信息表，结构如表 19.2 所示。创建表及相关字段的 SQL 语句如下：

```
SET SQL MODE="NO AUTO VALUE ON ZERO";
--
-- 数据库: 'treasure'
--
-- --------------------------------------------------------
--
-- 表的结构 'drawing'
--
CREATE TABLE 'drawing' (
  'id' int(11) NOT NULL auto increment,
  'realname' varchar(20) NOT NULL default '',
  'idcard' varchar(18) NOT NULL default '',
  'phone' varchar(20) NOT NULL default '',
  'address' varchar(50) NOT NULL default '',
  'zip' varchar(6) NOT NULL default '',
  'nickname' varchar(20) NOT NULL default '',
  'gname' varchar(50) NOT NULL default '',
  'status' tinyint(4) NOT NULL default '0',
  'submittime' datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY  ('id')
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

表 19.2　用户领奖信息表

字　　段	类　　型	说　　明
id	int(11)	记录 ID（自增）
realname	varchar(20)	用户真实姓名
idcard	varchar(18)	用户身份证号码
phone	varchar(20)	用户电话
address	varchar(50)	用户通讯地址
zip	varchar(6)	邮政编码
nickname	varchar(20)	昵称
gname	varchar(50)	获奖物品名称
status	tinyint(4)	状态
submittime	datetime	提交领奖单时间

用户领奖信息表用来记录用户领奖的信息的表。该表在实际的使用过程中，可以根据不同的业务类型调整相应的字段信息，同时也可以通过附表的方式扩展。

19.2.2　道具及奖品信息表

道具及奖品信息表用来记录游戏中道具、奖品情况的信息。在 treasure 数据库中建立一个命名为 gamegs 的道具及奖品信息表，结构如表 19.3 所示。创建表及相关字段的 SQL 语句如下：

```
SET SQL MODE="NO AUTO VALUE ON ZERO";
--
-- 数据库: 'treasure'
--
-- --------------------------------------------------------
--
-- 表的结构 'gamegs'
--
CREATE TABLE 'gamegs' (
  'id' int(11) NOT NULL auto increment,
  'gid' int(11) NOT NULL default '0',
  'location' mediumint(9) NOT NULL default '0',
  'count' tinyint(1) NOT NULL default '0',
  'owner' int(20) NOT NULL default '0',
  'locks' varchar(255) NOT NULL default '',
  'savetime' datetime NOT NULL default '0000-00-00 00:00:00',
  PRIMARY KEY  ('id'),
  KEY 'game id' ('id')
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=510 ;
```

表 19.3　道具及奖品信息表

字　　段	类　　型	说　　明
id	int(11)	记录 ID（自增）
gid	int(11)	物品 ID 编号
location	mediumint(9)	物品位置编号
count	tinyint(1)	物品数量（暂时没有用）

续表

字　段	类　型	说　明
Owner	int(20)	物品拥有者 ID
locks	varchar(255)	物品锁数量（以 PHP 序列化形式存储）
savetime	datetime	保存物品时间

道具及奖品信息表用来记录游戏中道具、奖品情况的信息。该表是用户与 Flash 交互的主要数据来源，该表记录物品的位置信息和物品锁数量，用户挖宝的过程就是与这张表信息交互的过程。

为了便于后面程序的调试，向数据库中插入两条测试数据，插入测试代码的 SQL 语句如下：

```
INSERT INTO 'gamegs' VALUES (491, 104, 187, 1, 0, '', '0000-00-00 00:00:00');
INSERT INTO 'gamegs' VALUES (2, 12, 35, 1, 0, '', '0000-00-00 00:00:00');
```

19.2.3　物品详细信息表

物品详细信息表用来记录物品的详细信息。在 treasure 数据库中建立一个命名为 goods 的物品详细信息表，结构如表 19.4 所示。创建表及相关字段的 SQL 语句如下：

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
--
-- 数据库: 'treasure'
--
-- --------------------------------------------------------
--
-- 表的结构 'goods'
--

CREATE TABLE 'goods' (
  'gid' int(11) NOT NULL default '0',
  'gname' varchar(30) NOT NULL default '',
  'gtype' tinyint(4) NOT NULL default '0',
  'content' varchar(255) NOT NULL default '',
  'pic' varchar(30) NOT NULL default '',
  'nmovep' int(11) NOT NULL default '0',
  'price' int(11) NOT NULL default '0',
  'utimes' int(11) NOT NULL default '0',
  PRIMARY KEY  ('gid'),
  KEY 'goods index id' ('gid'),
  KEY 'goods_gid' ('gid')
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

表 19.4　物品详细信息表

字　段	类　型	说　明
gid	int(11)	物品 ID 编号
gname	varchar(30)	物品名称
gtype	tinyint(4)	物品类型（0 表示消费物品，1 表示奖品）
content	varchar(255)	物品详细描述

续表

字　段	类　型	说　明
pic	varchar(30)	物品图片位置（暂时没有用）
nmovep	int(11)	消耗行动力数值
price	int(11)	物品价格（游戏币）
utimes	int(11)	物品可以使用次数

物品详细信息表用来记录物品的详细信息，由于这些物品都是在产品设计时已经确定的，所以该表的数据相当于一个基本的参数配置文件。使用表来存储，可以更便于业务升级和管理员使用。

由于程序需要初始化，所以向数据库中插入游戏物品初始化信息数据。插入代码的 SQL 语句如下：

```
INSERT INTO 'goods' VALUES (0, '木铲子', 0, '普通的挖宝工具,比较浪费体力','tm001',
10, 10, 30);
INSERT INTO 'goods' VALUES (1, '钢铲子', 0, '有了它估计能让您省很多力气', 'tm002',6, 200, 100);
INSERT INTO 'goods' VALUES (2, '金铲子', 0, '最昂贵也是最好的铲子', 'tm003', 4, 500, 999);
INSERT INTO 'goods' VALUES (12, '加蛋粉丝', 0, '增加120 点的体力', 'tm12', -120, 20, 1);
INSERT INTO 'goods' VALUES (14, '地雷', 0, '减少体力 100', '', 100, 10, 1);
INSERT INTO 'goods' VALUES (3, '小挂锁', 0, '小巧精美的锁头', 'tm03', 2, 1, 1);
INSERT INTO 'goods' VALUES (4, '钢丝锁', 0, '比较坚硬的锁头', 'tm04', 6, 5, 1);
INSERT INTO 'goods' VALUES (5, '加密锁', 0, '高科技的产物，想轻易的打开可不是一
件容易的事情', 'tm05', 10, 10, 1);
INSERT INTO 'goods' VALUES (6, '挂锁钥匙', 0, '专门用来打开小挂锁', 'tm06', 2, 1, 1);
INSERT INTO 'goods' VALUES (7, '钢丝钥匙', 0, '专门用来打开钢丝锁', 'tm07', 6, 5, 1);
INSERT INTO 'goods' VALUES (8, '加密钥匙', 0, '专门用来打开加密锁', 'tm08', 10, 10, 1);
INSERT INTO 'goods' VALUES (13, '炸弹', 0, '减少体力 100', 'tm13', 100, 5, 1);
INSERT INTO 'goods' VALUES (16, '探宝器', 0, '能够提示您地图上哪里有宝物,可使用
1 次', 'tm16', 0, 60, 1);
INSERT INTO 'goods' VALUES (10, '米粉丝', 0, '增加 50 点的体力', '', -50, 10, 1);
INSERT INTO 'goods' VALUES (11, '牛肉粉丝', 0, '增加250 点的体力', '', -250, 40, 1);
INSERT INTO 'goods' VALUES (15, 'TNT', 0, '减少体力 350', '', 350, 20, 1);
INSERT INTO 'goods' VALUES (17, '游戏币', 0, '给用户增加100 点的游戏币', '', 100, 0, 1);
INSERT INTO 'goods' VALUES (108, '宝贝不倒翁（蓝）', 1, '宝贝不倒翁（蓝）', '', 0, 10, 0);
INSERT INTO 'goods' VALUES (107, '宝贝不倒翁（粉）', 1, '宝贝不倒翁（粉）', '', 0, 10, 0);
INSERT INTO 'goods' VALUES (103, '剑锋传说', 1, '剑锋传说', '', 0, 50, 0);
INSERT INTO 'goods' VALUES (106, '丫丫玩偶', 1, '丫丫玩偶', '', 0, 10, 0);
INSERT INTO 'goods' VALUES (105, '可爱小熊', 1, '可爱小熊', '', 0, 10, 0);
INSERT INTO 'goods' VALUES (109, '皮皮熊', 1, '皮皮熊', '', 0, 20, 0);
INSERT INTO 'goods' VALUES (104, '老虎皮', 1, '老虎皮', '', 0, 100, 0);
INSERT INTO 'goods' VALUES (102, '液晶电视', 1, '液晶电视', '', 0, 800, 0);
INSERT INTO 'goods' VALUES (100, 'IDOL 木制音响', 1, 'IDOL 木制音响', '', 0, 120, 0);
INSERT INTO 'goods' VALUES (101, '方寸不乱', 1, '方寸不乱', '', 0, 300, 0);
```

19.2.4　用户资料信息表

用户资料信息表用来记录用户现在所拥有的信息。在 treasure 数据库中建立一个命名为 sessgs 的用户资料信息表，结构如表 19.5 所示。创建表及相关字段 SQL 语句如下：

```
SET SQL MODE="NO AUTO VALUE ON ZERO";
--
-- 数据库: 'treasure'
--
-- --------------------------------------------------------
--
-- 表的结构 'sessgs'
--

CREATE TABLE 'sessgs' (
  'uid' int(11) NOT NULL default '0',
  'gid' int(11) NOT NULL default '0',
  'gnums' int(11) NOT NULL default '0',
  'ltimes' int(11) NOT NULL default '0',
  'status' tinyint(4) NOT NULL default '0',
  PRIMARY KEY  ('uid', 'gid')
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

表 19.5　用户资料信息表

字　段	类　型	说　明
uid	int(11)	用户 ID
gid	int(11)	物品 ID
gnums	int(11)	物品数量
ltimes	int(11)	剩余使用次数
status	tinyint(4)	物品状态（1 表示正在使用的物品）

用户资料信息表用来记录用户现在所拥有的信息，其中重要的参数是物品数量、剩余使用次数和物品状态。当用户得到物品之后就通过该表与其他用户交互，完成游戏中宝贝争夺的业务逻辑。

为了便于后面程序的调试，向数据库中插入两条测试数据，插入测试代码的 SQL 语句如下：

```
INSERT INTO 'sessgs' VALUES (11, 0, 1, 30, 1);
INSERT INTO 'sessgs' VALUES (2, 0, 1, 13, 1);
```

19.2.5　游戏用户的信息表

游戏用户的信息表用来记录游戏用户的信息。在 treasure 数据库中建立一个命名为 session 的游戏用户的信息表，结构如表 19.6 所示。创建表及相关字段的 SQL 语句如下：

```
CREATE TABLE 'session' (
  'uid' int(11) NOT NULL auto increment,
  'sessionid' varchar(32) NOT NULL default '',
  'userid' varchar(20) NOT NULL default '',
  'username' varchar(20) NOT NULL default '',
  'level' tinyint(4) NOT NULL default '0',
  'mobile' varchar(11) NOT NULL default '',
  'movepoint' int(11) NOT NULL default '0',
  'score' int(11) NOT NULL default '0',
  'expiretime' datetime NOT NULL default '0000-00-00 00:00:00',
  'ipaddr' varchar(15) NOT NULL default '',
```

```
  PRIMARY KEY  ('uid')
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=15 ;
```

表 19.6　游戏用户的信息表

字　段	类　型	说　明
uid	int(11)	游戏用户 ID 编号（自增）
sessionid	varchar(32)	SessionID
userid	varchar(20)	用户 ID
username	varchar(20)	用户昵称
level	tinyint(4)	用户等级（暂时没有用）
mobile	varchar(11)	用户手机号码
movepoint	int(11)	用户行动力点数
score	int(11)	游戏币数量
expiretime	datetime	SessionID 过期时间
ipaddr	varchar(15)	用户 IP 地址

游戏用户的信息表用来记录游戏用户的信息，记录用户在游戏的数据交换，如用户行动力点数和游戏币数量。同时，当用户消耗行动力点数和游戏币数量后，可以通过手机号码进行在线的充值和兑换。

为了便于后面程序的调试，向数据库中插入两条测试数据，插入测试代码的 SQL 语句如下：

```
INSERT  INTO  'session'  VALUES  (4,  '81b86f558df4381814912b6d12189a92',
'1089949', '咚咚锵', 0, '', 500, 5757, '2007-08-14 11:05:47', '192.168.
10.184');
INSERT INTO 'session' VALUES (5, '8a700068ef819e52f80c1ff0fd710457', '128',
'随风飘流', 0, '', 500, 33, '2007-10-11 16:05:39', '192.168.10.187');
```

19.3　基础设计及说明

本节将为读者介绍程序分类描述、文件及目录功能描述、API 封装包说明、程序返回信息代码说明，以上这些共同构成藏宝游戏模块的基础设计的整体架构。读者通过本节的学习，会对藏宝游戏模块有一个整体的认识。

19.3.1　程序分类描述

藏宝游戏整个程序主要由挖宝游戏、后台管理（游戏管理、奖品管理）、前台用户程序（奖品领取、充值兑换）3 个部分组成。

1．挖宝游戏

挖宝游戏是通过 PHP 脚本程序驱动 Flash 文件，通过改变动态的获得动作类型和参数实现挖宝的游戏功能。在程序实现方面通过向 test.php 程序传值，程序通过接收 otype 来判

断进行操作的类型，然后调用对应的 API 封包完成该指定的操作。

2．后台管理

后台管理部分包括游戏管理和奖品管理。

游戏管理对应的文件为 manage_drawing.php，包含的功能为添加、删除、编辑等功能，对应的功能处理文件为 manage_process.php 及 manage_dprocess.php。

奖品管理对应的文件为 manage_drawing.php，主要实现了对获奖用户的一些基本的管理功能，在不同的活动中可以对这部分做细节的调整。

3．前台用户程序

前台用户程序主要涉及用户在游戏中的互动信息，比如奖品领取、游戏充值和积分兑换等。这部分内容在之前的模块中已经详细讲解过了，相信读者对这类的应用也一定不会陌生，可以尝试独立完成这部分的编码实现，这里不作为本章的重点说明。

19.3.2　文件及目录功能描述

前面介绍程序的功能分类和整个程序的组成部分，下面对具体的文件及目录的具体功能解释和说明。读者通过文件和目录的功能描述，可以直观地了解各个程序之间是如何相互配合共同完成业务逻辑的。

程序所有文件放置在根目录文件夹中，具体文件功能如下。

（1）核心程序说明。

- a.swf：游戏程序（Flash 格式）。
- alltool.php：获取所有道具。
- checkchongzhi.php：充值成功检测（Ajax 调用文件）。
- checkinput.php：用户输入信息验证（Ajax 调用文件）。
- delexpireuser.php：删除过期用户。
- drawingprize.php：用户领取奖品。
- drawing_process.php：领取奖品处理。
- flashgame.php：游戏主程序。
- manage_dprocess.php：奖品管理的相关处理。
- manage_drawing.php：奖品管理页面。
- manage_game.php：游戏管理页面。
- manage_process.php：游戏管理的相关处理。
- objectSwap.js：Flash 调用的 JS。
- pay.php：充值页面。
- pay_start.php：充值分步程序。
- pay_2.php：充值分步程序。
- pay_final.php：充值分步程序。
- test.php：Flash 与 PHP 进行交互的文件。

（2）class（文件夹）功能：类库文件夹。

- class.commonfg.php：普通用户操作类（统计、管理）。
- class.drawing.php：用户领奖类。
- class.mysql.php：数据库类。
- class.template.php：模板类。
- class.treasure.php：用户挖宝类。

（3）images（文件夹）功能：图片存放文件夹。

（4）inc（文件夹）功能：配置文件存放文件夹。

- inc.config.php：普通配置文件。
- inc.dbconfig.php：数据库配置文件。
- inc.function.php：程序中的通用函数。

（5）js（文件夹）功能：JavaScript 程序存放文件夹。

- appleClass.js：Ajax 简易对象。
- common.js：各个页面用到的 JS 函数集合。
- drawing.js：领奖页面用到的 JS 函数集合。

（6）style（文件夹）功能：样式表存放文件夹。

（7）tmp（文件夹）功能：页面缓存文件。

（8）tpl（文件夹）功能：页面模板文件。

- flashgame.tpl：挖宝程序首页模板。
- manage_drawing.tpl：获得奖品用户列表。
- manage_game.tpl：后台管理程序模板。

19.3.3　API 封装包说明

API（Application Programming Interface，应用程序编程接口）是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件的访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。在藏宝游戏模块的开发中，将不同的动作封装成独立的 API 接口，完成一项具体的功能，这样就可以直接调用某个接口或者组合使用多个接口程序，共同完成某一个业务逻辑。

1．查询某一个位置内容

【功能描述】 用户点击地图的某一个格子以进行挖宝操作，系统返回该格子所拥有的物品信息。

注意：每进行一次操作都要消耗行动力的值，根据使用的道具不一样（木铲子、钢铲子、金铲子）消耗的行动力也不一样。

【方法】 GetContentByLocation($pos, $sSid, $type=0)

```
//@param pos          地图位置（1-200）
//@param sSid         sessionid（用户会话 ID）
```

【功能】 查询地图指定位置包含的物品。

【传输数据（Flash 方面需要传递的参数）】 用户名（uname）、用户 ID（uid）、地图位置（pos）、操作方式（otype='local'）。

【PHP 方面需要返回的参数】 特殊格式的物品的信息。

【成功返回】 物品信息、物品类型（普通物品，使用道具）、用户行动力。

格式：r=正确标识&v=行动力值&s=游戏币值&w=加锁情况&wp=游戏物品编号|物品编号|物品名称|位置。

例如：

```
r=1&v=320&s=9350&w=3&wp=1|100|《挑战》周星驰签名 T 恤|1
```

【失败返回】

格式：r=错误标识&eno=错误代号&v=行动力值&s=游戏币值

例如：

```
r=0&eno=-6&ec=没有任何物品&v=340&s=9350
```

【服务器端的操作动作】

（1）判断 session 是否有效，否则返回失败。

（2）判断挖宝道具是否还可以使用，否则返回失败。

（3）判断是否有足够的行动力，否则返回失败。

（4）扣减 user 的行动力值（movepoint）。

（5）成功返回：a.没有任何物品 b.返回的是普通物品 c.返回的是道具。

2．查询用户信息

【功能描述】 获得用户的各项信息；

【方法】 GetSessionInfo($sSid)

```
//@param uname  用户名字
```

成功返回用户的信息，否则返回错误代码。

格式：r=错误标识&eno=错误代号&v=行动力值&s=游戏币值

例如：

```
r=0&eno=-6&ec=没有任何物品&v=340&s=9350
```

3．物品移动

【功能描述】 用户通过点击出来的物品的类型，如果为非道具（奖品）就可以移动该物品到其他的格子中以改变它的位置。

【方法】 MoveGoods($sSid, $posF, $posD, $ggid, $gid)

```
//@param $sSid       sessionid（会话 ID）
//@param $posF       物品原位置（1-200）
//@param $posD       物品目的位置（1-200）
//@param $ggid       游戏物品编号
//@param $gid        物品编号
```

【传输数据（Flash 方面需要传递的参数）】 用户名（uname）、用户 ID（uid）、物品原位置（posF）、物品目标位置（posD）、游戏物品编号（ggid）、物品编号（gid）、操作类型（otype='move'）。

【PHP 方面需要返回的参数】

格式：r=标识&eno=代号&v=行动力值&s=游戏币值

```
例如：r=1&eno=4&ec=移动成功&v=340&s=9350
```

【服务器端操作动作】

（1）判断 SessionID 是否有效，否则返回失败。

（2）判断物品是否在原位置，否则返回失败。

（3）判断是否有足够的行动力，否则返回失败。

（4）行动力的减少规则根据用户使用挖宝道具的情况来确定。

（5）如果移动用户不是当前拥有者，判断物品锁定是否有锁，否则返回失败，更新物品位置到 b，更新结束时间为当前时间+10 分钟，更新当前物品拥有者。

（6）如果移动用户是当前拥有者，更新物品位置到 b，更新物品锁定为 0。

4．保存物品

【功能描述】 用户通过单击“保存”按钮可以对移动到指定位置的物品进行保存，也就是把宝物的所有者设置为当前的用户。

【方法】 UpdateGoodsOwner($ggid, $uid)

```
//@param $ggid       游戏物品编号
//@param $uid        用户 ID
```

【传输数据（Flash 方面需要传递的参数）】 用户名（uname）、用户 ID（uid）、游戏物品编号（ggid）、操作方式（otype='save'）。

【PHP 方面需要返回的参数】

格式：r=标识&eno=代号&v=行动力值&s=游戏币值

例如：

```
r=1&eno=4&ec=移动成功&v=340&s=9350
```

【服务器端操作动作】 从数据库中更改指定记录的 owner。

5．物品加锁

【功能描述】 护宝的用户可以对自己挖宝的宝物进行加锁操作，这样如果其他用户找到了这个物品必须解锁才可以移动或者占有。

【方法】 AddLocks($uname, $ggid, $locktype)

```
//@param $uname          用户名
//@param $ggid           游戏物品编号
//@param $locktype       道具编号
```

【传输数据（Flash 方面需要传递的参数）】 用户名（uname）、用户 ID（uid）、游戏物品编号（ggid）、锁类型（locktype）操作方式（otype='add'）。

【PHP 方面需要返回的参数】

格式：r=标识&eno=代号&v=行动力值&s=游戏币值

例如：

```
r=1&eno=4&ec=移动成功&v=340&s=9350
```

【服务器端操作动作】

（1）判断道具编号是否为锁类型，否则返回失败。

（2）判断用户是否拥有这个道具，否则返回失败。

（3）判断是否有足够的行动力，否则返回失败。

（4）扣减 user 的行动力值（movepoint）。

（5）通过数据库给指定游戏编号的记录添加加锁信息，同时对用户道具列表进行更新。

6．物品解锁

【功能描述】 每一个物品都有可能被物品拥有者对其进行加锁，其他用户如果也想要得到这个物品，必须用道具（相对应的钥匙）对其解锁。

【方法】 OpenLocks($sSid, $uname, $ggid, $locktype, $keytype)

```
//@param      $sSid          sessionid（会话 ID）
//@param      $uname         用户名
//@param      $ggid          游戏物品编号
//@param      $locktype      道具类型（锁）
//@param      $keytype       道具类型（钥匙）
```

【传递数据（Flash 方面需要传递的参数）】 用户名（uname）、用户 ID（uid）、游戏物品编号（ggid）、锁类型（locktype）、钥匙类型（keytype）、操作方式（otype=‘open’）。

【PHP 方面需要返回的参数】

格式：r=标识&eno=代号&v=行动力值&s=游戏币值

例如：

```
r=1&eno=4&ec=移动成功&v=340&s=9350
```

【服务器端操作动作】

（1）判断道具编号是否为解锁类型，否则返回失败。

（2）判断是否有足够的行动力，否则返回失败。

（3）扣减 user 的行动力值（movepoint）。

（4）判断使用该道具的用户是否拥有该道具，否则返回失败。

（5）利用道具编号和用户 ID 进行道具清除（删除该记录）。

（6）对解锁物品进行解锁操作，同时对用户道具列表进行更新。

7．检测用户所拥有的物品是否被其他用户移动

【功能描述】 用户在找到宝物并且护宝期间，会每间隔一个时间段查询一次自己即将拥有的物品是否被移动了，如果物品被移动了返回给用户提示信息。

【方法】 hasMoveGoods($pos, $ggid)

```
//@param $pos    地图位置（1-200）
//@param $ggid  游戏物品编号
```

【传递数据（Flash 方面需要传递的参数）】 用户名（username）、用户 ID（uid）、地图位置（pos）、游戏物品编号（ggid）、操作类型（otype='check'）。

【PHP 方面需要返回的参数】

格式：r=标识&eno=代号&v=行动力值&s=游戏币值

例如：

```
r=1&eno=4&ec=移动成功&v=340&s=9350
```

【服务器端操作动作】 判断用户所拥有的物品是否在指定位置并给出相应信息。

8．重置游戏

【功能描述】 当用户在护宝期间点击重置游戏按钮时，对游戏进行重置（主要情况分为：护宝状态和非护宝状态）。

【方法】 ResetGame($ggid)

```
//@param $ggid        游戏物品编号
```

【传递数据（Flash 方面需要传递的参数）】用户名（username）、用户 ID（uid）、游戏物品编号（ggid）、操作类型（otype='reset'）。

9．获得用户自身道具库

【功能描述】 Flash 会通过参数传递来获得用户自身所拥有的道具。

【方法】 GetUserTool($sSid,$uid="")

```
//@param      $sSid        用户会话 ID(SessionID)
//@param      $uid         用户 ID（用户 ID）
```

【传递数据（Flash 方面需要传递的参数）】 用户名（username）、用户 ID（uid）、操作类型（otype='tool'）。

【PHP 方面需要返回的参数】返回 XML 格式数据信息。

格式：r=标识&eno=代号&v=行动力值&s=游戏币值

例如：

```
r=1&eno=4&ec=移动成功&v=340&s=9350
```

10．道具使用

【功能描述】 看到前面的物品描述可以知道，物品分为两种：道具和奖品。道具可以使用并且有特殊的效果。

【方法】 UserAppointTool($toolno="",&$p)

```
//@param $toolno     道具编号
//@param &$p       指向数组的指针，这个数组中包含用户名、游戏物品编号以及所有道具类型
```

【传递数据（Flash 方面需要传递的参数）】用户名（username）、用户 ID（uid）、

道具编号（toono）操作类型（otype='use'）。

【PHP 方面需要返回的参数】

格式：r=标识&eno=代号&v=行动力值&s=游戏币值

例如：

```
r=1&eno=4&ec=移动成功&v=340&s=9350
```

【服务器端操作动作】

（1）判断道具编号是否符合要求，否则给出错误信息。

（2）判断用户是否拥有指定的道具，否则给出错误信息。

（3）根据道具编号来划分应该执行哪一个程序分支。

11. 购买道具

【功能描述】 道具有的时候可以挖到，但是有一些道具还是需要通过购买来获得。

【方法】 BuyAppointTool($toolno)

```
//@param $toolno          道具编号
```

【传递数据（Flash 方面需要传递的参数）】用户名（username）、用户 ID（uid）、道具编号（toono）操作类型（otype='buy'）。

【服务器操作动作】

（1）检查是否存在这个道具，否则返回错误信息。

（2）获得用户的游戏币数量。

（3）判断用户游戏币数量是否可以购买道具，否则返回错误信息。

（4）更新用户游戏币数量，返回购买成功信息。

19.3.4　程序返回信息代码说明

在前面介绍了 API 封包的功能和具体的服务器动作。由于 API 接口的特点使用者无法知道具体的操作状态，所以在程序设计之初就规定了程序的返回值和对应的含义。具体返回值的含义如表 19.7 所示。

表 19.7　程序返回值说明

返回值	说　明	返回值	说　明
100	您已经得到该物品	–11	加锁失败
5	道具购买成功	–12	用户没有这个道具
4	加锁成功	–13	这个道具不是锁类型道具
3	移动成功	–14	物品没有上锁，无需解锁
2	保存成功	–15	钥匙不能打开这把锁，钥匙与锁不匹配
1	操作成功	–16	您没有开锁的钥匙
0	操作失败	–17	这个道具不是钥匙
–1	SESSIONID 无效	–18	您的积分不够
–2	没有足够的行动力	–19	没有这个道具

续表

返回值	说　明	返回值	说　明
–3	系统故障	–20	查询失败
–4	物品不在指定位置	–21	您不是该物品的拥有者
–5	SESSIONID 冲突	–22	您没有选择要使用的道具
–6	没有任何物品	–23	非用户使用道具
–7	物品已经没有锁	–24	用户只能够在自己所拥有的物品上进行加锁
–8	物品被其他用户移动了	–100	用户尚未登录
–9	物品已经被别人占有	default	数据库操作失败
–10	没有可使用挖宝道具		

注意：函数根据编号返回一个关联数组。

```
return array("flag"=>$flag, "errno"=>$fno, "errdesc"=>$errDesc);
```

其中，flag 为 1 表示执行成功的操作，为 0 表示执行失败的操作；errno 表示编号；errdesc 表示提示信息的语言描述。

19.4　后台管理程序

后台管理程序是负责对游戏中的数据进行管理（比如编辑、添加等一些基本的管理操作）。但是与通常的管理程序不同的是，藏宝游戏的后台管理程序是与前台的 Flash 游戏直接关联的，可以通过后台的操作影响到用户的游戏结果。

藏宝游戏模块的后台管理程序，按照功能可以划分为：

- 管理信息显示程序；
- 管理操作程序；
- 增加物品程序。

下面分别介绍这 3 个部分的程序实现，并对涉及的技术细节做详细说明。

19.4.1　管理显示程序

管理后台程序中。使用频率最高的就是管理页面的显示程序。该程序将当前游戏的物品信息和坐标参数以列表的形式提供给管理员，方便管理员了解物品的投放情况。而且可以通过搜索框的方式选择显示“全部信息”、“道具信息”和“奖品信息”。

创建文件名为 manage_game.php 的 PHP 脚本文件，程序通过读取指定接口获得当前的游戏信息数据。该文件的具体代码如下：

```
<?php
//说明：游戏管理程序
define ( 'ROOT_PATH', dirname ( __FILE__ ) . '/' );
//加载库文件
require_once (ROOT_PATH . 'inc/inc.dbconfig.php');
```

```
require_once (ROOT_PATH . 'inc/inc.config.php');
require_once (ROOT_PATH . 'inc/inc.function.php');
require_once (ROOT_PATH . 'class/class.mysql.php');
require_once (ROOT_PATH . 'class/class.treasure.php');
require_once (ROOT_PATH . 'class/class.commonfg.php');
require_once (ROOT_PATH . 'class/class.template.php');
$url = $_SERVER ["HTTP_HOST"] . $_SERVER ["PHP_SELF"]; //获得服务器路径信息
$arrUserID = array ("828067", "11212" );                //用户 id
$search = var_process ( "get", "search", "all" );      //接收搜索结果值
$currpage = var_process ( "get", "page", 1 );
$itemnum = 10;                                          //显示条数
//实例化数据库类
$oMysql = new MYSQL ( "", $db_conf ["host"], $db_conf ["user"], $db_conf
["pass"], $db_conf ["db"] );
$oCommonFG = new CommonFG ( $oMysql );
$tpl = new Template ();                                 //实例化模板类
//获取当前搜索对应的个数信息
$count = $oCommonFG->Count_UP_List ( $search );
$page_info = divpage_toadmin ( $count, $currpage, $itemnum ); //翻页信息
//获得物品用户游戏列表
$game_info = $oCommonFG->Get_UP_List ( $itemnum, $page_info ["currpage"],
$search );
//获得物品列表
$good_info = $oCommonFG->GetGoodsInfo ();
if ($game_info !== false) {
    for($i = 0; $i < count ( $game_info ); $i ++) {
        $gid = intval ( $game_info [$i] ["gid"] );
        $ownerid = intval ( $game_info [$i] ["owner"] );
        $locks = $game_info [$i] ["locks"];

        $tmp = $oCommonFG->GetGnameByGID ( $gid );
        $game_info [$i] ["gname"] = $tmp ["gname"];
        if ($ownerid != 0) {                            //判断物品所有人
            $tmp = $oCommonFG->GetUnameByUID ( $ownerid );
            $game_info [$i] ["ownern"] = $tmp ["username"];
        }
        if (strlen ( $locks ) != 0) {               //判断物品加锁情况
            $tmp = unserialize ( $locks );
            $index = 0;
            foreach ( $tmp as $key => $value ) {    //循环获取信息，生成数组
                $game_info [$i] ["locksinfo"] [$index] ["id"] = $key;
                $game_info [$i] ["locksinfo"] [$index] ["name"] = $locks
                name [$key];
                $game_info [$i] ["locksinfo"] [$index] ["nums"] = $value;
                $index ++;
            }
        }
    }
} else {
//省略相关操作
}
//返回数组
$page_url = array ();
$base_url = "manage_game.php";
$stype_get = empty ( $search ) ? "" : "search=" . $search . "&";
//翻页参数设置
$page_url ["first"] = $base_url . "?" . $stype_get . "page=" . $page_info
["firstpage"];
$page_url ["pre"] = $base_url . "?" . $stype_get . "page=" . $page_info
```

```
["prepage"];
$page_url ["next"] = $base_url . "?" . $stype_get . "page=" . $page_info
["nextpage"];
$page_url ["end"] = $base_url . "?" . $stype_get . "page=" . $page_info
["max_page"];
//Smarty 内容标签
$tpl->assign ( "gameinfo", $game_info );
$tpl->assign ( "goodinfo", $good_info );
$tpl->assign ( "pageinfo", $page_info );
$tpl->assign ( "pageurl", $page_url );
$tpl->assign ( "stype", $search );
//加载管理模板页面
$tpl->display ( "manage_game.tpl" );
?>
```

【代码解读】

上面程序的核心逻辑由 3 个步骤组成：

（1）获得匹配到的信息个数。

（2）获得游戏用户的物品列表、获得物品和道具的详细资料。

（3）将这些数据生成模板标签，输出到页面模板中。

程序描述如下：

```
//获取当前搜索对应的个数信息
$count = $oCommonFG->Count_UP_List($search);
$page_info = divpage_toadmin($count, $currpage, $itemnum);
//获得物品用户游戏列表
$game_info = $oCommonFG->Get_UP_List($itemnum, $page_info["currpage"],
$search);
//获得物品列表
$good_info = $oCommonFG->GetGoodsInfo();
```

游戏管理页面 HTML 模板（manage_game.tpl）部分主要涉及的是数据的逻辑展示，读者仔细体会模板引擎将数据信息循环展示的标签代码。该文件核心的代码如下：

```
<table border="0" cellpadding="5" cellspacing="0" id="info">
    <tr>
        <td colspan="7">
        <form action="manage_game.php" method="get"><select
            name="search">
            <option value="all" {|if $stype=="all"|}selected{|/if|}>全部信
            息</option>
            <option value="tool" {|if $stype=="tool"|}selected{|/if|}>道
            具信息</option>
            <option value="prize" {|if $stype=="prize"|}selected{|/if|}>
            奖品信息</option>
        </select> <input type="submit" value="搜索" /></form>
        </td>
    </tr>
    <tr class="head">
        <th>ID</th>
        <th>物品名称</th>
        <th>地图位置</th>
        <th>拥有者</th>
        <th>物品加锁情况</th>
```

```
        <th>保存时间</th>
        <th>操作</th>
    </tr>
    {|section loop=$gameinfo name=num|}
    <tr onMouseOver="Switch TR Bg Color(this,'over');"
        onMouseOut="Switch TR Bg Color(this,'out');">
        <td>{|$gameinfo[num].id|}</td>
        <td>{|$gameinfo[num].gname|}</td>
        <td>{|$gameinfo[num].location|}</td>
        <td>{|if $gameinfo[num].owner == 0|}   {|else|}
        {|$gameinfo[num].ownern|} {|/if|}</td>
        <td>{|if $gameinfo[num].locks == ""|}   {|else|} {|section
        loop=$gameinfo[num].locksinfo name=index|}
        {|$gameinfo[num].locksinfo[index].name|}:{|$gameinfo[num].
        locksinfo[index].nums|}   
        {|/section|} {|/if|}</td>
        <td>{|if $gameinfo[num].savetime == "0000-00-00 00:00:00"|}
          {|else|} {|$gameinfo[num].savetime|} {|/if|}</td>
        <td><a href="#editform"
            onClick="ChangeForm('edit',{|$gameinfo[num].id|},{|$gameinfo
            [num].gid|},{|$gameinfo[num].location|});">修改</a>  
        <a onClick="return CusConfirm('确认是否删除此记录');"
            href="manage_process.php?type=del&id={|$gameinfo[num].id|}
            &stype={|$stype|}">删除</a>  
        <a
            href="manage process.php?type=qedit&id={|$gameinfo[num].id|}
            &stype={|$stype|}">位置快速更新</a>
        {|if $gameinfo[num].gid >= 100|} <a href="#locks"
            onClick="ChangeForm('locks',{|if $gameinfo[num].locks == "
            "|}''{|else|}'{|section loop=$gameinfo[num].locksinfo
            name=index|}{|$gameinfo[num].locksinfo[index].id|}{|$game-
            info[num].locksinfo[index].nums|}{|/section|}'{|/if|},{|$ga-
            meinfo[num].id|})">加/解锁</a>
        {|/if|}</td>
    </tr>
    {|/section|}
    <tr>
        <td colspan="7"> </td>
    </tr>
    <tr>
        <th colspan="7" style="color: #AAAAAA">10 条记录/页  
        共{|$pageinfo.max_page|}页  
        第{|$pageinfo.currpage|}页      
                 
        {|if $pageinfo.max page == 1|} [首页] [上一页] [下一页] [尾页] {|else|} {|if
        $pageinfo.currpage == 1|} [首页] [上一页] <a href="{|$pageurl.next|}">
        [下一页]</a>
        <a href="{|$pageurl.end|}">[尾页]</a> {|elseif $pageinfo.currpage ==
        $pageinfo.max_page|} <a href="{|$pageurl.first|}">[首页]</a> <a
            href="{|$pageurl.pre|}">[上一页]</a> [下一页] [尾页] {|else|} <a
            href="{|$pageurl.first|}">[首页]</a> <a href="{|$pageurl.pre|}">
            [上一页]</a>
        <a href="{|$pageurl.next|}">[下一页]</a> <a href="{|$pageurl.end|}">
        [尾页]</a>
```

```
        {|/if|} {|/if|}</th>
    </tr>
    <tr>
        <td colspan="7"><input type="button" class="add_btn"
            value="添加游戏信息记录" onClick="ChangeForm('add')" /></td>
    </tr>
</table>
```

在浏览器中执行下面的链接：

```
http://localhost/newfg/manage_game.php
```

藏宝游戏管理页面的浏览效果，如图 19.1 所示。



图 19.1　藏宝游戏管理页面

当用户在游戏中获得物品后，管理员可以通过后台了解到物品的详细信息，同时提供了加锁和解锁的选项链接。

当管理员在后台选择了“奖品信息”或者在浏览器中执行下面的链接：

```
http://localhost/newfg/manage_game.php?search=prize
```

游戏奖品信息管理页面的浏览效果，如图 19.2 所示。

19.4.2　管理操作程序

藏宝游戏是一个互动性很强的游戏，因此需要管理人员随时通过游戏管理后台改变游戏里的物品信息和基本设置。管理后台提供给管理人员的主要管理功能有物品修改、删除、更新位置、加\解锁。其中，修改操作、加锁\解锁操作通过 manage_game.php 程序来完成，增加物品操作、删除和快速更新位置由 manage_process.php 程序来完成。下面分别介绍这

几个管理功能的编码实现和基本原理。



图 19.2　游戏奖品信息管理页面

1．修改操作

物品修改操作是管理最常用的功能之一。该功能可以快速更改物品和所在的位置，这个功能在藏宝游戏中使用频率非常高。

由于使用了类的封装，因此该操作的业务逻辑、代码非常简单，只有两条语句：

```
//获得匹配结果的个数
$count = $oCommonFG->Count_UP_List($search);
//通过获得的参数取得物品的详细信息
$game_info = $oCommonFG->Get_UP_List($itemnum, $page_info["currpage"], $search);
```

【代码解读】

在获得该条详细信息后会提交负责展示的页面模板，模板中的 JavaScript 程序通过判断提交的动作参数，将确定需要生成表格的结构。该程序代码如下：

```
function ChangeForm(type) {
    var oAddForm = document.getElementById("addform");
    var oEditForm = document.getElementById("editform");
    var oLocksForm = document.getElementById("locksform");
    <!--判断操作类型-->
    if(type == "add"){                  <!--增加动作-->
        oAddForm.style.display = "";
        oEditForm.style.display = "none";
        oLocksForm.style.display = "none";
    }else if(type == "edit") {          <!--编辑动作-->
        var id =  arguments[1];
        var gid = arguments[2];
        var location = arguments[3];
        var oSelGid = document.getElementById("egid");
```

```
        var oSelLocation = document.getElementById("elocation");
        var oID = document.getElementById("ggid");
        var oLabelID = document.getElementById("edit id");
        if(oLabelID.innerHTML != "") {      <!--判断返回元素标签的 HTML-->
            oLabelID.innerHTML = "";
        }
        oLabelID.innerHTML = parseInt(id);  <!--返回标签赋值 HTML -->
        oID.value = id;
        oAddForm.style.display = "none";
        oEditForm.style.display = "";
        oLocksForm.style.display = "none";
        <!--生成复选框数组-->
        for(var i=0; i<oSelGid.options.length; i++) {
                        if(parseInt(oSelGid.options[i].value) == parseInt(gid)) {
                            oSelGid.options[i].selected = true;
                break;
                        }
                }
        for(var i=0; i<oSelLocation.options.length; i++) { <!--生成操作复选框-->
            if(oSelLocation.options[i].value == location) {
                oSelLocation.options[i].selected = true;
                break;
            }
        }
    }
    else if(type == "locks") {                          <!--锁定状态-->
        var sLocksInfo = arguments[1];
        var id =  arguments[2];
        var iLast = 0;
        var iIndex = 0;
        var oSomeLock;
        var oID = document.getElementById("locks ggid");

        var oLabelID = document.getElementById("locks id");
                if(oLabelID.innerHTML != "") {
                        oLabelID.innerHTML = "";
                }
                oLabelID.innerHTML = parseInt(id);<!--返回标签赋值 HTML -->
        oID.value = id;
                oAddForm.style.display = "none";
                oEditForm.style.display = "none";
                oLocksForm.style.display = "";
        if(sLocksInfo.length != 0) {                   <!--如果存在加锁信息-->
            var aLocksInfo = sLocksInfo.split("");
            for(var i=0; i<aLocksInfo.length; i+=2) {<!--生成操作复选框数组-->
                iLast = i+1;
                iIndex = parseInt(aLocksInfo[i]) + 1;
                oSomeLock = document.getElementById("lock " + iIndex);
                iOptions = oSomeLock.length;
                for(var j=0; j<iOptions; j++) {
                    if(oSomeLock.options[j].value == aLocksInfo[iLast])
                        oSomeLock.options[j].selected = true;
                }
```

```
            }
        }
    }
setTimeout("Hidden_OTB('"+type+"')", 20000); <!--在 20000 毫秒数后调用函数-->
}
```

【代码解读】

确定了插入代码标签的位置后，通过下面的函数修改表格的 style.display 的属性，实现表格内容在页面上的显示。该 JavaScript 函数如下：

```
function Hidden_OTB(form_type) {
    var oForm = document.getElementById(form_type+"form");
    oForm.style.display = "none";<!--设置元素为隐藏属性-->
}
```

在游戏管理页面中选择需要修改的物品，单击“修改”链接，游戏物品或奖品的修改页面效果如图 19.3 所示。

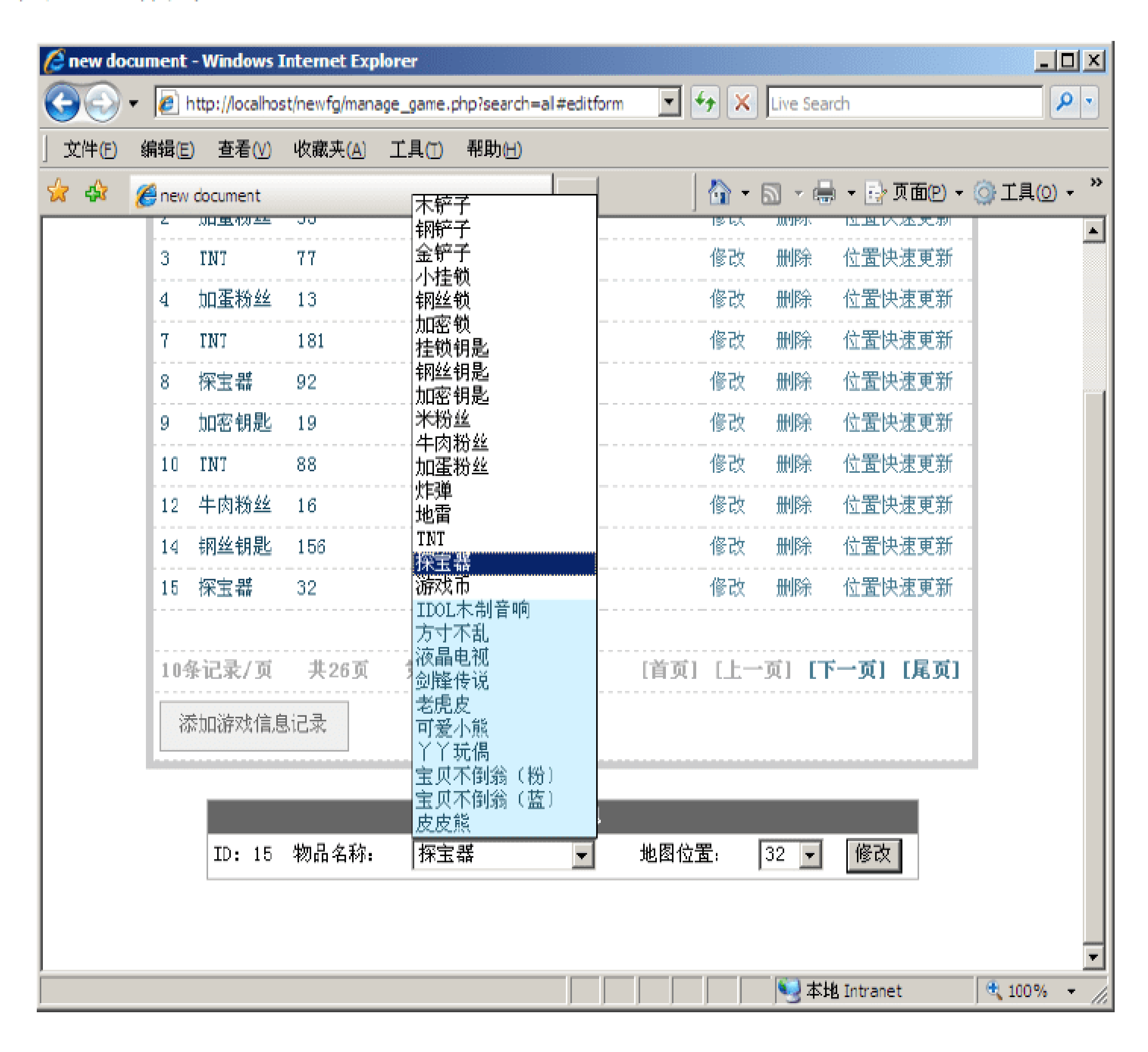


图 19.3　游戏物品修改

2．加锁及解锁操作

管理员可以通过管理后台了解用户获得的宝物的情况，后台提供进行加锁操作，这样如果其他用户找到了这个物品，必须解锁才可以移动或者占有，解锁部分的业务也是同样的，这里不再赘述。

前面介绍过 API 的封包，这里直接使用接口程序实现加锁功能。加锁方法如下：

```
AddLocks($uname, $ggid, $locktype)
//@param $uname          用户名
//@param $ggid           游戏物品编号
//@param $locktype       道具编号
```

编码时调用 AddLocks 这个方法，同时提供用户名参数、游戏物品编号参数、道具参数编号就可以完成奖品加锁的操作了。

解锁的过程和加锁相同，调用解锁的方法名即可。方法名及参数说明如下：

```
方法：OpenLocks($sSid, $uname, $ggid, $locktype, $keytype)
//@param     $sSid       sessionid（会话 ID）
//@param     $uname      用户名
//@param     $ggid       游戏物品编号
//@param     $locktype   道具类型（锁）
//@param     $keytype    道具类型（钥匙）
```

【代码解读】

编码时调用 OpenLocks 这个方法，同时提供 sessionid（会话 ID）参数、用户名参数、游戏物品编号参数、道具类型（锁）、道具类型（钥匙）就可以完成对加锁奖品的解锁操作了。

管理员对用户物品的加锁页面如图 19.4 所示。

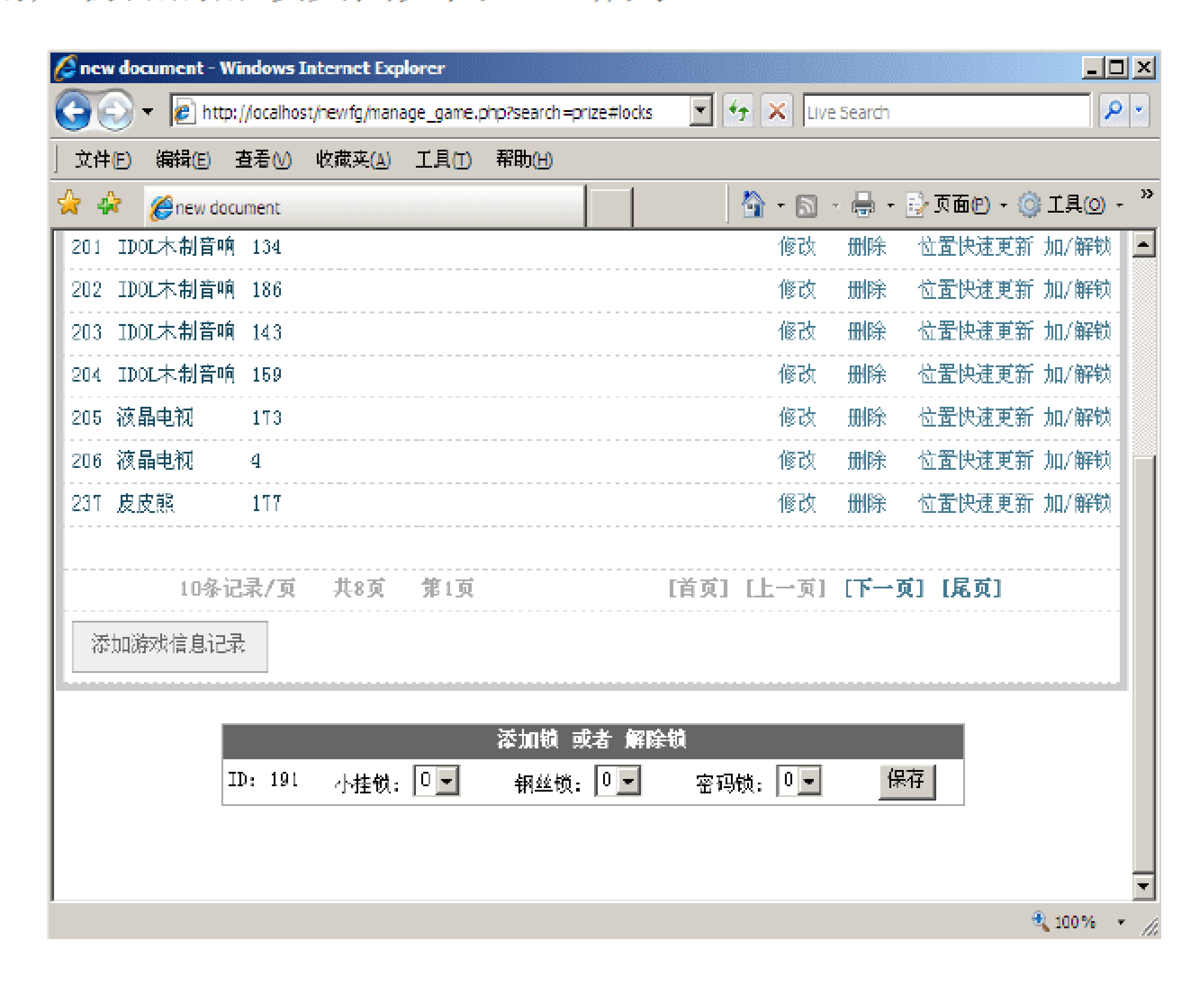


图 19.4　获奖物品加锁

3．增加物品操作

增加物品功能是一个基础功能，游戏中的所有物品都是通过这个功能实现的。这个功能由两个步骤组成：

（1）选定一个添加的物品；

（2）为该物品在游戏地图中指定一个位置。

在之前介绍过增加物品等操作都是由 manage_process.php 这个文件完成的。该文件的代码如下：

```
<?php
define ( 'ROOT_PATH', dirname ( __FILE__ ) . '/' );
//加载库文件
require_once (ROOT_PATH . 'inc/inc.dbconfig.php');
```

```
require once (ROOT PATH . 'inc/inc.config.php');
require once (ROOT PATH . 'inc/inc.function.php');
require once (ROOT PATH . 'class/class.mysql.php');
require once (ROOT PATH . 'class/class.treasure.php');
require once (ROOT PATH . 'class/class.commonfg.php');
//获得匹配条件类型
if (GP_Method ( "post" )) {                    //获得参数的方法 POST
    $type = var process ( "post", "type", "" );
    $stype = var process ( "post", "stype", "all" );
    $id = var process ( "post", "id", "" );
    $gid = var_process ( "post", "gid", "" );
    $location = var process ( "post", "location", "" );
    $page = var process ( "post", "page", 1 );
    } else {                                   //Get 方法获得参数
    $type = var_process ( "get", "type", "" );
    $stype = var process ( "get", "stype", "all" );
    $id = var process ( "get", "id", "" );
    $gid = var process ( "get", "gid", "" );
    $location = var process ( "get", "location", "" );
    $page = var process ( "get", "page", 1 );
}
//创建数据库链接
$oMysql = new MYSQL ( "", $db conf ["host"], $db conf ["user"], $db conf
["pass"], $db conf ["db"] );
$oCommonFG = new CommonFG ( $oMysql );
//判断操作类型
if ($type == "del") {                              //删除操作
    $is ok = $oCommonFG->Delete_UPByID ( $id );
} else if ($type == "edit") {                      //编辑操作
    //print r($ POST);
    $is ok = $oCommonFG->Edit UPByID ( $id, $gid, $location );
} else if ($type == "qedit") {                     //快速更新操作
    $is ok = $oCommonFG->Quick UPByID ( $id );
} else if ($type == "add") {                       //新增操作
    $is ok = $oCommonFG->ADD UP ( $gid, $location );
} else if ($type == "locks") {                     //加锁操作
    $locks [] = intval ( var process ( "post", "lock 1", 0 ) );
    $locks [] = intval ( var process ( "post", "lock 2", 0 ) );
    $locks [] = intval ( var process ( "post", "lock 3", 0 ) );
    $locks tmp = "";
    for($i = 0; $i < count ( $locks ); $i ++) {
        if ($locks [$i] != 0) {
            $locks tmp [$i] = $locks [$i];
        }
    }
    $locks info = empty ( $locks tmp ) ? "" : serialize ( $locks tmp );
    //获得加锁信息
    $is ok = $oCommonFG->Edit UP Locks ( $id, $locks info );
}
//流程控制
if ($is ok) {
    if (empty ( $stype )) {
```

```
        header ( "Location:manage_game.php" );          //浏览器重定向
    } else {
        header ( "Location:manage_game.php?search=" . $stype . "&page=" . $page );
    }
} else {
    echo "<a href='manage_game.php'>更新速度太快系统出错，点击返回上一页面</a>";
}
?>
```

【代码解读】

在上面的程序中，通过判断$type 的类型来实现业务流程的控制。对应的增加物品的业务流程代码如下：

```
else if($type == "add") {   //新增操作
    $is_ok = $oCommonFG->ADD_UP($gid,$location);
}
```

当$type 变量的值为 add 时，调用 ADD_UP($gid,$location);方法，该方法的代码如下：

```
function ADD_UP($gid,$location) {
    $sql = "INSERT INTO gamegs('gid', 'location') VALUES(" . intval($gid) .
    "," . intval($location) . ")";      //执行插入 SQL 语句
    $this->db->query($sql);
    return $this->db->affected_rows() ? true : false;
}
```

【代码解读】

上面的 ADD_UP($gid,$location)方法具体的操作是向数据库的物品表中的游戏道具、奖品表中插入一条数据。

在游戏管理页面中点击“增加游戏信息记录”标签，可以看到增加物品的页面，如图 19.5 所示。

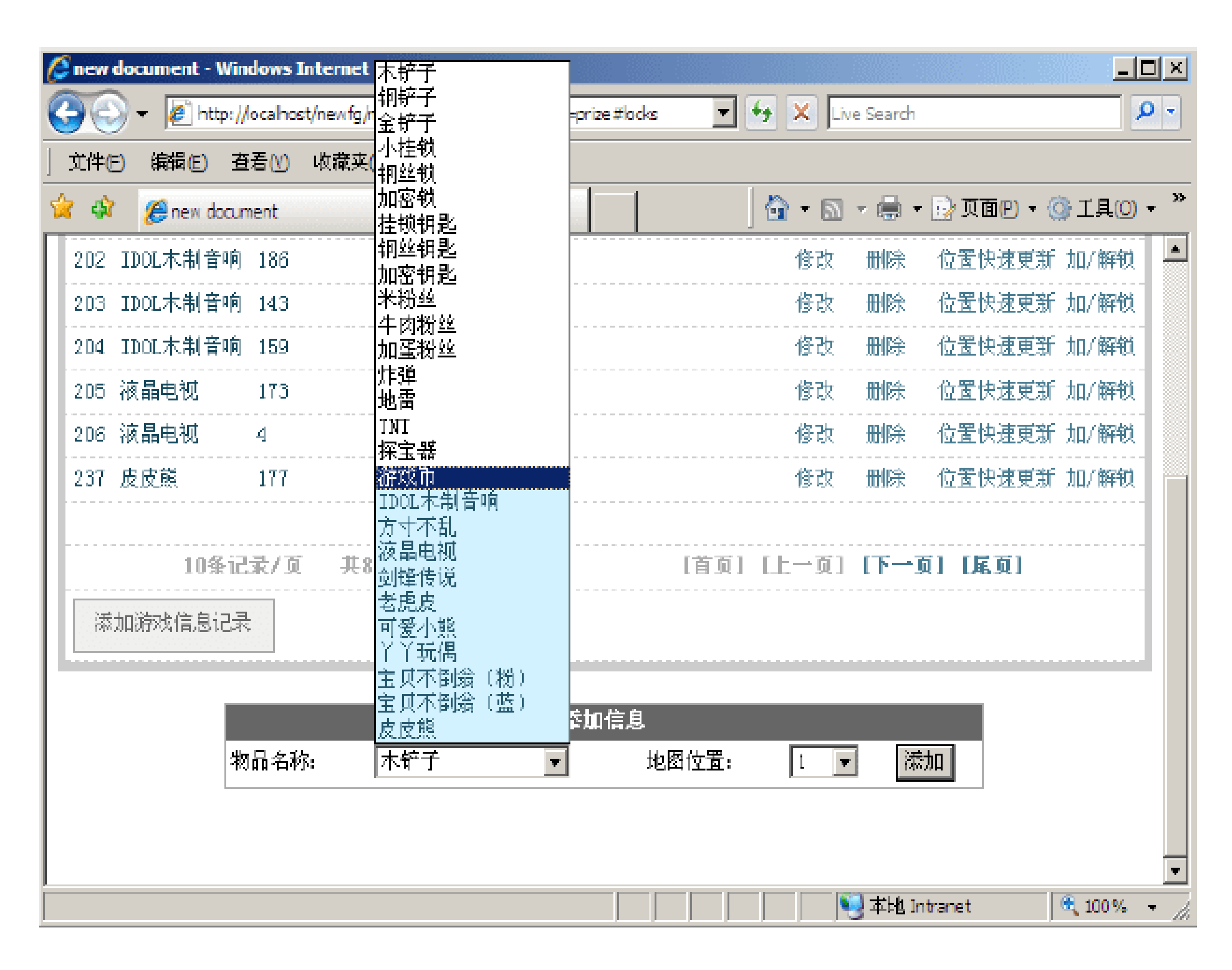


图 19.5　增加物品操作

4．删除操作

通过之前的介绍可以知道，物品的删除操作也是在 manage_process.php 中完成的，对

应的物品删除操作的流程代码如下：

```
if($type == "del") {                                    //判断操作类型
    $is_ok = $oCommonFG->Delete_UPByID($id);            //删除物品
}
```

当$type 变量的值为 del 时，调用 Delete_UPByID($id);方法。该方法的代码如下：

```
/* 删除游戏记录 */
function Delete_UPByID($id) {
    if(!empty($id)) {                                       //判断记录是否存在
        $sql = "DELETE FROM 'gamegs' WHERE 'id'=" . $id;
        $this->db->query($sql);
        return $this->db->affected_rows() ? true : false;  //返回操作结果
    }
    else {
        return false;
    }
}
```

物品的删除操作方法将数据库中指定的 id 的记录删除，即完成了指定物品的删除操作功能。

在游戏管理页面中选择需要删除的物品，单击“删除”链接，可以看到提示物品删除的页面，如图 19.6 所示。



图 19.6　物品删除提示

5．更新位置操作

更新位置的操作也是在 manage_process.php 中完成的，该操作的作用是将该物品随机

移动到地图的另一个位置。对应更新位置的操作流程代码如下：

```
//快速更新操作
else if($type == "qedit") {
    $is ok = $oCommonFG->Quick UPByID($id);
}
```

当$type 变量的值为 qedit 时，调用 Quick_UPByID($id);方法，该方法的代码如下：

```
/* 通过 id 更改地图位置 */
function Quick_UPByID($id) {
    if(!empty($id)) {
        $rand_num = mt_rand(1,200);                    //获得随机数
        $sql = "UPDATE 'gamegs' SET 'savetime'='0000-00-00 00:00:00',
        owner='', 'location'=" . $rand num . " WHERE 'id'=" . intval($id);
        $this->db->query($sql);
        return $this->db->affected rows() ? true : false;  //返回操作结果
    }
    else {
        return false;
    }
}
```

【代码解读】

该函数首先随机在 1～200 中取得一个数，然后在通过 id 更新数据库中对应记录物品位置信息的该条记录，完成更新物品位置信息的功能。

19.5　藏宝游戏程序

藏宝游戏程序负责游戏功能的实现，藏宝游戏是基于数据库驱动的 PHP 与 Flash 文件交互的应用。Flash 文件实现了游戏的素材和定义动作的接口及参数，PHP 通过动态的获得数据库中的信息，然后生成游戏参数发送给 Flash 文件，通过以上步骤完成藏宝游戏的业务流程。

藏宝游戏程序包含两个部分，即挖宝游戏程序和游戏用户信息列表程序。

下面分别介绍这两个程序的编码实现。

19.5.1　挖宝游戏程序

挖宝程序是基于数据库驱动的 PHP 与 Flash 交互的程序应用，挖宝游戏程序分为入口程序（负责加载 Flash 框架和处理页面模板）和游戏驱动程序。

1. 入口程序

入口程序的代码：

```
<?php
//功能：藏宝游戏入口程序
define ( "ROOT PATH", dirname (  FILE  ) . "/" );
//加载函数库
```

```
require once (ROOT PATH . "class/class.template.php");
require once (ROOT PATH . 'inc/inc.dbconfig.php');
require once (ROOT PATH . 'inc/inc.config.php');
require_once (ROOT_PATH . 'inc/inc.function.php');
require once (ROOT PATH . 'class/class.mysql.php');
require once (ROOT PATH . 'class/class.treasure.php');
require once (ROOT PATH . 'class/class.commonfg.php');
mb internal encoding ( "GB2312" );      //设置字符集为 GB2312（简体中文）
$max_string = 3;                        //设置字符长度
$oMysql = new MYSQL ( "", $db conf ["host"], $db conf ["user"], $db conf
["pass"], $db conf ["db"] );
$statis = new CommonFG ( $oMysql );     //实例化公共操作类
if (1 !== 1) {                          //判断用户身份
    $uid = $userobj->getUserId ();
    $user = $userobj->getUserInfo ();
    $uname = $user ["nickname"];
} else {                                //“游客”身份登录
    $uid = 828067;
    $uname = "游客";
}
if ($uid == 0) {
    header
("Location:http://localhost/newfg/flashgame.phpbackurl=http://localhost
/newfg/flashgame.php" );
}
$highScore = '';
$highScore = $statis->StatHighScore ();        //获得最高分数
$obtainUG = '';
$obtainUG = $statis->StatObtainUG ();
//遍历该用户所属的物品
for($i = 0; $i < count ( $obtainUG ); $i ++) {
    if (mb strlen ( $obtainUG [$i] [1] ) > $max string) {
        $obtainUG [$i] [] = mb_substr ( $obtainUG [$i] [1], 0, $max_string ) . "..";
} else {
        $obtainUG [$i] [] = $obtainUG [$i] [1];
    }
}
//生成模板标签
$tpl->assign ( "uid", $uid );
$tpl->assign ( "uname", $uname );
$tpl->assign ( "ginfo", $obtainUG );
$tpl->assign ( "hscore", $highScore );
//加载模板页面
$tpl->display ( "flashgame.tpl" );
?>
```

【代码解读】

文件入口程序的核心作用是设定各种基础的信息（比如用户身份信息等），然后将页面模板中涉及的数据生成 Smarty 的标签，再加载对应的模板文件。该 HTML 模板（flashgame.tpl）页面代码如下：

```
<div id="content">
    <div id="mainbg">
        <div id="right">
            <div id="score">
            {|section loop=$hscore name=num|}
            <div class="rec">
            <label class="uscore">{|$hscore[num].username|}</label><label
```

```
            class="sscore">{|$hscore[num].score|}</label>
            </div>
            {|/section|}
            </div>
            <div id="gotgoods">
                <div id="yanhy">
                    <div id="yanhy1">
                    <ul>
                    <li id="start">获宝名单如下:</li>
                    {|section loop=$ginfo name=num|}
                    <li title="{|$ginfo[num][0]|}  获得{|$ginfo[num][1]|}">
                    {|$ginfo[num][0]|}  :{|$ginfo[num][2]|}</li>
                <li> </li>
                    {|/section|}
                    <li id="end">恭喜以上获奖用户</li>
                    </ul>
                    </div>
                    <div id="yanhy2"> </div>
                    <div id="yanhy3"> </div>
                </div>
            </div>
        </div>
        <div id="left">
            <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
            id="flvplay" codebase="http://fpdownload.macromedia.com/pub/
            shockwave/cabs/flash/swflash.cab#ve
rsion=8,0,0,0" width="833" height="663" align="middle">
            <param name="allowScriptAccess" value="sameDomain" />
            <param name="movie" value="a.swf?username={|$uname|}&uuid={|$uid|}" />
            <param name="quality" value="high" />
            <param name="allowFullScreen" value="true" />
                                <embed src="a.swf?username={|$uname|}&uuid=
                                {|$uid|}" id="ffplay" quality="high" width=
                                "833"  height="663" name="flvplayer" align=
                                "middle"allowscriptaccess="sameDomain"
                                allowFullScreen="true" type="application/
                                x-sho-ckwave-flash" pluginspage="http://www.
                                macromedia.com/go/getflashplayer" />
                        </object>
        </div>
    </div>
</div>
```

【代码解读】

上面代码中<object></object>中的内容就是 Flash 游戏文件，给定相关的参数如下：

```
<embed src="a.swf?username={|$uname|}&uuid={|$uid|}" id="ffplay" quality
="high" width="833" height="663" name="flvplayer" align="middle" allowscr-
iptaccess="sameDomain" allowFullScreen="true" type="application/x-shock-
waveflash" pluginspage="http://www.macromedia.com/go/getflashplayer" />
```

在浏览器中执行 http://localhost/newfg/flashgame.php 链接，预览如图 19.7 所示。

2．游戏驱动程序

挖宝游戏驱动程序主要涉及 test.php 文件和 class.treasure.php 文件，每进行一次游戏操作（游戏初始化），Flash 都要调用 test.php（该文件中定义了 treasure 类的实例）。驱动器

文件 test.php 根据要操作的类型执行相应的动作，并返回某种格式的信息给 Flash。该驱动器文件（test.php）的代码如下：



图 19.7　初始化 Flash 挖宝游戏

```
<?php
define('ROOT_PATH', dirname(__FILE__) . '/');
//加载公用函数库
require_once(ROOT_PATH . 'inc/inc.dbconfig.php');
require_once(ROOT_PATH . 'inc/inc.config.php');
require_once(ROOT_PATH . 'inc/inc.function.php');
require_once(ROOT_PATH . 'class/class.mysql.php');
require_once(ROOT_PATH . 'class/class.treasure.php');
session_start();                                    //初始化 Session
$method = $_SERVER["REQUEST_METHOD"];               //获得当前方法的接收类型
if(strtoupper($method) == "GET" )                   //判断是否为 GET 模式
{
    //GET 模式下参数完整性校验
    $uname = !empty($_GET["uname"]) ? $_GET["uname"] : "";
    $uid = !empty($_GET["uid"]) ? $_GET["uid"] : "0";
    $otype = !empty($_GET["otype"]) ? $_GET["otype"] : "";
    $pos = !empty($_GET["pos"]) ? $_GET["pos"] : "";
    $posF = !empty($_GET["pos"]) ? $_GET["pos"] : "";
    $posD = !empty($_GET["pos2"]) ? $_GET["pos2"] : "";
    $ggid = !empty($_GET["ggid"]) ? $_GET["ggid"] : 0;
    $gid = !empty($_GET["gid"]) ? $_GET["gid"] : "";
    $locktype = !empty($_GET["locktype"]) ?  $_GET["locktype"] : "";
    $keytype = !empty($_GET["keytype"]) ? $_GET["keytype"] : "";
    $p = $_GET;
}
else if(strtoupper($method) == "POST")              //判断是否为 POST 模式
{
    //POST 模式下参数完整性校验
    $uname = !empty($_POST["uname"]) ? $_POST["uname"] : "";
    $uid = !empty($_POST["uid"]) ? $_POST["uid"] : "0";
    $otype = !empty($_POST["otype"]) ? $_POST["otype"] : "";
    $pos = !empty($_POST["pos"]) ? $_POST["pos"] : "";
    $posF = !empty($_POST["pos"]) ? $_POST["pos"] : "";
```

```
    $posD = !empty($_POST["pos2"]) ? $_POST["pos2"] : 1;
    $ggid = !empty($_POST["ggid"]) ? $_POST["ggid"] : 0;
    $gid = !empty($_POST["gid"]) ? $_POST["gid"] : "";
    $locktype = !empty($_POST["locktype"]) ? $_POST["locktype"] : "";
    $keytype = !empty($_POST["keytype"]) ? $_POST["keytype"] : "";
    $p = $_POST;
}
```

【代码解读】

上述代码主要是在“业务流程”执行之前完成基础准备工作，具体的内容是：

（1）相关类库文件加载。

（2）Session 的初始化。

（3）当前参数接收模式判断（POST、GET）。

（4）游戏基本参数完整性校验。

在完成基础准备工作后，下面介绍通过 Switch()语句控制业务流程，该部分代码将判断用户当前的操作“内容”。具体的代码如下：

```
$uname = !empty($uname) ? $uname : "";             //获得用户名（三元算子）
$oMysql = new MYSQL("", $db_conf["host"], $db_conf["user"], $db_conf["pass"],
$db_conf["db"]);
$oSession = new Treasure($oMysql, $uname, $uid);   //实例化挖宝业务类
if(($errorCode = $oSession->GetErrorCode()) > 0)   //获得异常错误信息
{
    //当前操作目标（流程控制）
    switch($otype)
    {
    //查询地图指定位置的情况
    case "local":
            $info = $oSession->GetContentByLocation($pos, session_id(), $otype);
      break;
    //保存物品
    case "save":
            $info = $oSession->UpdateGoodsOwner($ggid);
            break;
    //物品移动
    case "move":
            $info = $oSession->MoveGoods(session_id(), $posF, $posD, $ggid, $gid);
            break;
    //使用道具
    case "use":
            $info = $oSession->UserAppointTool($gid,$p);
            break;
    //解锁操作
    case "open":
            $info = $oSession->OpenLocks(session_id(), $uname, $ggid,
            $locktype, $keytype);
            break;
    //加锁操作
    case "add":
            $info = $oSession->AddLocks($uname, $ggid, $locktype);
      break;
    //物品保存期间检测物品是否被其他用户移动
    case "check":
            $info = $oSession->hasMoveGoods($pos, $ggid);
            break;
    //游戏复位
```

```
    case "reset":
            $info = $oSession->ResetGame($ggid);
            break;
    //当前用户拥有道具
    case "tool":
            $info = $oSession->GetUserTool($oSession->GetSessionID(), $uid);
            break;
    //购买道具
    case "buy":
            $info = $oSession->BuyAppointTool($gid);
            break;
    //自动抢夺用户护宝物品
    case "auto":
            $info = $oSession->AutoPlunderTreasure();
            break;
    //成功获得道具
    case "success":
            $info = $oSession->DefineGoodsSuccess($ggid,$gid);
            break;
    //获得全部道具
    case "alltool":
            $info = $oSession->GetAllToolInfo();
            //print r($info);exit;
            break;
    default:
            echo session id();
            exit;
    }
}
else
{
    $info = $oSession->ErrorInfo($errorCode);
}
```

【代码解读】

在游戏引擎（test.php）程序中，程序通过接收 otype 来判断进行哪种操作。otype 可以为如下值。

- local：查询地图指定位置的情况。
- move：物品移动。
- save：保存物品。
- use：使用道具。
- open：解锁操作。
- add：加锁操作。
- check：物品保存期间检测物品是否被其他用户移动。
- reset：游戏复位。
- tool：当前用户拥有道具。
- buy：购买道具。
- success：成功获得道具。
- alltool：获得全部道具。

注意：otype 值只是说明要完成什么样的操作，具体还要配合其他参数，详见 API 封装包说明。

通过上述的流程控制，引擎程序已经执行了用户的操作请求（数据库业务模型）。由于是引擎程序与向浏览器输出内容的模式不同，所以在下面将获得操作信息（指令）转换字符集并按照规定的格式输出，代码如下：

```
$session_id = $oSession->GetSessionID();           //获得当前的 SessionID
$result_si = $oSession->GetSessionInfo($session_id);  //获取用户信息
//工具类型判断
if(($otype == "tool" || $otype == "alltool") && $errorCode != -5)
{
    header("Content-Type:text/xml");                //声明文件类型
    if(isset($info['flag'])) {
        echo iconv("GB2312", "UTF-8", ResultFormat($info,0,0));
                                                    //字符集转换
    }
    else  {
        echo ResultFormatXML($info);                //输出格式化结果
    }
}
else if($otype == "check") {         //物品保存期间检测物品是否被其他用户移动
    if(isset($info['flag'])) {echo iconv("GB2312", "UTF-8", ResultFormat
    ($info,$result_si["movepoint"],
$result_si["score"]));
    }
    else {
        list($lnums,$hnums) = $info;echo "r=1&ls=$lnums&hs=$hnums" . '&v=' .
        $result_si["movepoint"] . '&s=' .
$result_si["score"];
    }
}
else if(($otype == "use" && $gid == 16) || (count($info) <= 4))//使用道具
{
    header("Content-Type:text/xml");                    //声明文件类型
    if(isset($info['flag'])) {echo iconv("GB2312", "UTF-8", ResultFormat
    ($info,$result_si["movepoint"],
    $result_si["score"]));
    }
    else {
        $str = implode(",", $info);                     //组合信息结果
        $str .= '&v=' . $result_si["movepoint"] . '&s=' . $result_si
        ["score"];
        echo 'cc=' . $str;
    }
}
else
{
    header("Content-Type:text/html");           //向浏览器声明文件类型
    echo iconv("GB2312", "UTF-8", ResultFormat($info,$result_si["move-
    point"],$result_si["score"]));
}

    $oMysql->database_close();                  //关闭数据库连接
session_destroy();                              //销毁 Session
?>
```

【代码解读】

（1）在游戏主页面的 tpl 文件（flashgame.tpl）中<object>元素中给出的 Flash 地址要跟随两个参数，即：

```
a.swf?username=yhbsww&uuid=828067;
```

（2）测试程序的时候由于 Flash 是根据操作以 POST 方式传递参数，如果需要查看结果可以在每执行一次操作的时候按 Ctrl+*（小键盘）快捷键，再一次按*屏蔽提示信息。

另外，可以在浏览器中直接输入 test.php 后面，加上 GET 方式的参数也可以进行单步测试，例如：

```
test.php?otype=use&gid=1&uname=yhbsww&uid=828067;
```

在游戏地图选择任意位置开始挖宝，游戏预览如图 19.8 所示。



图 19.8　挖宝游戏界面

19.5.2　游戏用户信息程序

游戏用户信息的程序主要功能是显示获宝用户名单列表和财富爬行榜列表。下面具体介绍下这两个列表的编码实现。

1．获宝用户名单列表

获宝用户名单列表是在 flashgame.php 这个程序入口文件中实现的。展示获宝用户名单列表的代码如下：

```
$obtainUG ='';
$obtainUG = $statis->StatObtainUG();              //用户对应的物品
for($i=0; $i<count($obtainUG); $i++) {
    if(mb_strlen($obtainUG[$i][1]) > $max_string) {    //字符串截取
        $obtainUG[$i][] = mb_substr($obtainUG[$i][1], 0, $max_string) . "..";
    }
    else {
        $obtainUG[$i][] = $obtainUG[$i][1];
    }
}
```

```
$tpl->assign("ginfo", $obtainUG);                          //奖品详情标签
```

获宝用户名单列表是在 flashgame.php 程序对应的模板部分，作用是格式化奖品详细标签（以列表的形式展示）。模板标签展现代码如下：

```
<div id="yanhy1">
<ul>
    <li id="start">获宝名单如下:</li>
    {|section loop=$ginfo name=num|}
    <li title="{|$ginfo[num][0]||}  获得  {|$ginfo[num][1]||}">
    {|$ginfo[num][0]||}  :{|$ginfo[num][2]||}</li>
    <li> </li>
    {|/section|}
    <li id="end">恭喜以上获奖用户</li>
</ul>
</div>
```

获宝用户名单的页面预览如图 19.9 所示。

2．财富爬行榜列表

财富爬行榜列表也是在 flashgame.php 这个程序入口文件中实现的，财富爬行榜列表的代码如下：

```
$highScore ='';
$highScore = $statis->StatHighScore();//获得最高分记录
$tpl->assign("hscore", $highScore);
```

对应的模板部分的标签展现逻辑如下：

```
<div id="score">
{|section loop=$hscore name=num|}
<div class="rec">
<label class="uscore">{|$hscore[num].username|}</label>
<label class="sscore">
{|$hscore[num].score|}</label>
</div>
{|/section|}
</div>
```

财富爬行榜列表的页面预览如图 19.10 所示。



图 19.9　获宝用户名单



图 19.10　财富爬行榜列表

19.6　小　　结

藏宝游戏模块是典型的数据库驱动的 PHP 与 Flash 交互的游戏应用程序。本章从藏宝游戏概述开始讲起，分别介绍了藏宝游戏说明、游戏道具描述。第 19.1 节主要是明确游戏要完成目标和需要开发的功能。第 19.2 节开始数据库的设计，该节由 5 个基本表组成。第 19.3 节介绍了基础设计部分，有程序分类描述、文件及目录功能描述、API 封装包说明和程序返回信息代码说明。在第 19.4 节着手搭建后台管理程序，管理员可以通过后台来实现对游戏的管理。最后介绍藏宝游戏程序的设计和编码实现，藏宝游戏由两个部分组成，即挖宝游戏程序和游戏用户信息程序。

第 20 章　图片交易平台（MVC 设计模式）

本章将为读者详细讲述如何设计并实现一个图片交易平台模块。图片交易平台模块是典型的平台应用程序，它通过对后台图片数据和信息的有效管理来支撑前端的应用，具体来说就是图片交易的业务流程。读者通过本章的学习可以了解到，如何通过需求分析设计和规划平台系统；在具体的技术层面，可以了解到 MVC 的程序开发模式。该模式也是作为本章的要点着重为读者介绍，目前很多流行程序框架都应用 MVC 的设计模式进行代码的组织。

全面地理解 MVC 的设计模式，可以更好地理解开源框架，同时对提高自身的编程能力有很大帮助。技术方面的另一个重点就是图片交易流程的程序实现，这其中的要点是“购物车”程序的设计和开发。由于本章讲述的内容属于应用系统的开发，因此涉及众多功能和应用，读者可以在学习的过程中对之前介绍过的相关知识点进行复习和梳理。

本章主要涉及的知识点如下。

- MVC 的概念：模型－视图－控制器。MVC 并不是 PHP 这类 B/S 结构程序中的概念，它本来是应用于 Desktop 程序中的一种软件设计模式，M 是指数据模型，V 是指用户界面，C 则是控制器。
- E-R 关系图：数据库设计的一种方法。该方法把每个表看成是一个实体，实体之间的关系（E-R）反映了两个实体之间的逻辑关系，而 E-R 关系图是通过图形化的方式显示两个实体的关系。
- 构造函数：构造函数是一种特殊的方法，用来创建对象时初始化对象，通常用作为成员变量赋值。

20.1　系 统 概 述

图片交易平台是以艺术品数据库的数据支持为基础的平台应用，设计目标是面向艺术出版社开发设计数字图片交易平台。该平台整合、挖掘了出版社数字图片资源，实现艺术出版资源的综合共享、检索、查询、调用。

整个图片交易平台系统是依托艺术品数据库，通过电子商务的模式结合传统的线下订单交易、配送及人工电话订购服务，开展面向艺术品、图书出版、图片交易服务。

20.1.1　基础架构设计

前面提到图片交易平台是以艺术品数据库为依托的，这是一个国家级的艺术品类的专

用数据库，图片交易平台是这个数据库的一个具体的应用。因此设计的首要任务就是设计数据库同步的方案和流程。

同步的过程通过程序系统接口完成，主要通过以下两类工具实现：

1．中转工具

具体包括文件监控、XML 解析、转换出库申请单。

2．同步出库工具

具体包括状态监控、文字信息出库、网络图片出库。

图片交易平台数据同步流程如图 20.1 所示。

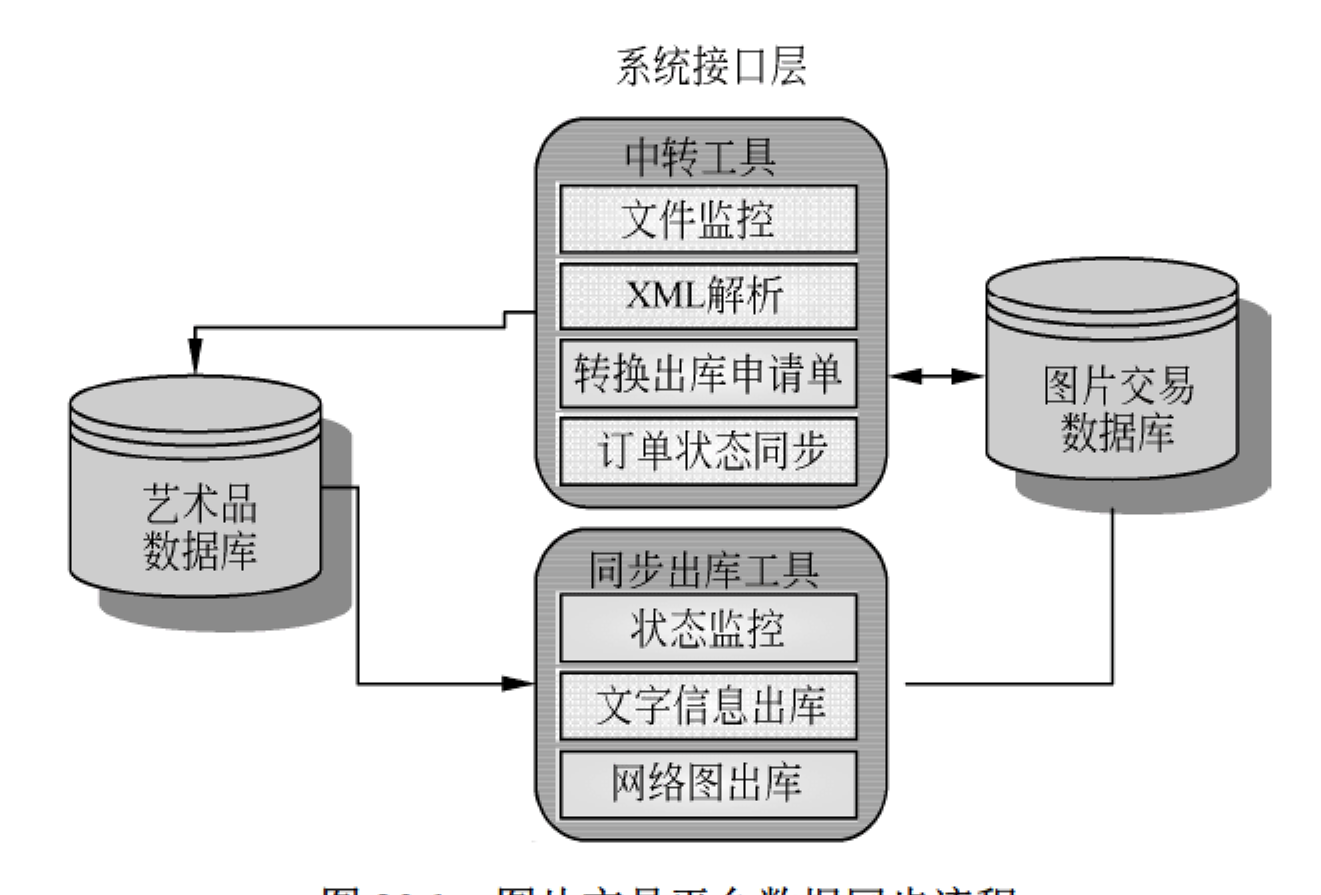


图 20.1　图片交易平台数据同步流程

在了解了数据获取的流程之后，下面介绍图片交易平台的整体框架构成。整个流程包括 4 个部分：

- 艺术品数据库同步；
- 图片交易平台数据库数据入库；
- 图片交易平台后台管理；
- 图片交易平台前端程序。

以上的 4 个部分构成了图片交易平台的模块的基本框架。图片交易平台整体架构如图 20.2 所示。

图片交易平台的业务流程是整个平台的业务主线，各项功能和服务都是围绕着这条主线进行的。

核心的业务就是图片的交易，该业务包含了以下的步骤：

（1）会员登录。

（2）图片搜索。

（3）生成订单。

（4）订单处理（记录）。

（5）图片出库。

（6）确认交易。

图片交易平台业务流程图如图 20.3 所示。

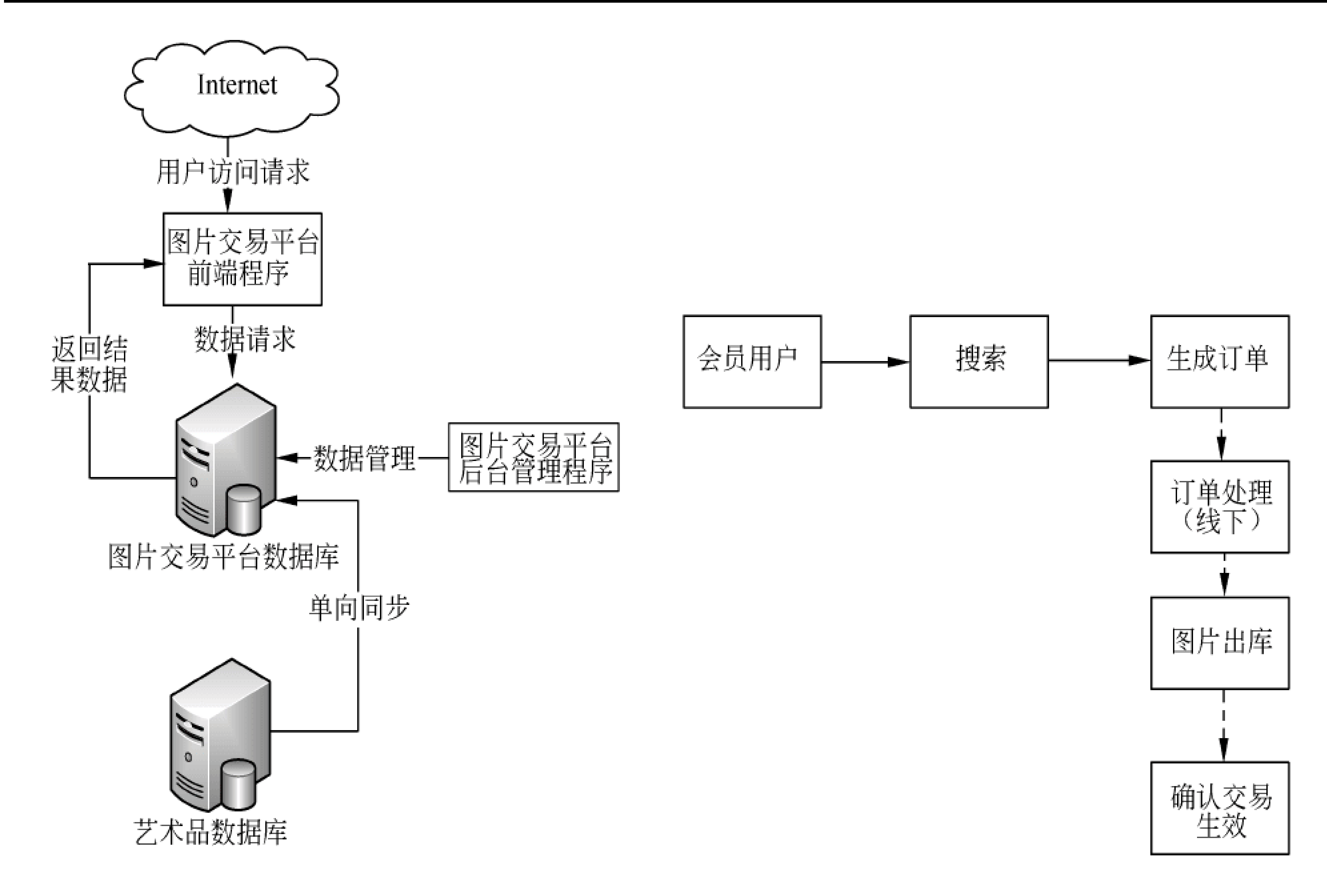


图 20.2　图片交易平台整体架构业务流程　　　　图 20.3　图片交易平台业务流程图

20.1.2　核心功能描述

图片交易平台可以分为前端和后台管理两个部分。下面分别介绍这两个部分需要实现的主要功能。

图片交易平台前端的主要功能描述如表 20.1 所示。

表 20.1　图片交易平台前端功能表

编号	功　能	实 现 方 式
1	会员认证	系统对外开放，普通用户可以直接浏览检索，但是不能使用购物车和提交订单；只有会员才可以使用购物车和提交订单；会员不能自行注册，只能由系统统一分配
2	集成门户	集成门户的展示。可以展示专题，推荐图片和资讯
3	艺术品导航浏览	系统将提供按照艺术品分类导航，按艺术家分类导航和名称缩写导航等导航方式，供用户快速浏览最近入库的相关艺术品图片，对于历史资料通过全文检索方式获取。对于其他资源不再提供导航功能，通过检索结果进行浏览
4	艺术品概览	系统对艺术品，艺术家提供三种展示方式：图片列表、图文列表、文字列表三种展示方式。具有幻灯片展示功能
5	艺术品详细信息展示	在图片详情展示上与内部发布平台不同，出版交易平台上展示的内容将更侧重于其出版业务需求，同时对版权归属将重点展示。具体包括尺寸、大小、开本、格式、名称、作者、版权归属、版权许可范围、材质（如反转片、电子片），出版记录。艺术品只显示与艺术品图片本身有关的属性及关联信息，但不显示其关联的艺术家等的详细信息

续表

编号	功　　能	实 现 方 式
6	关键字检索	指可以根据作品名、作者名等关键字对艺术品进行检索
7	复杂检索	指可以根据版权类别等详细信息对艺术品进行更全面、有针对性的检索
8	辅助检索功能	关联关键词检索：根据此图的关键词搜索带有此关键词的其他图片 热点搜索词：统计并展示数据库检索的热点搜索词。 在结果中检索：提供在结果中的渐进检索
9	购物车	要购买的图片可以先放入购物车内，购物车中的内容包括缩略图、时间、价格。对于购物车内的条目可以进行增加、删除
10	订单管理	提供订单的相关功能如下： 发起订单 修改订单 提交订单 订单打印 订单生效 审核授权与回收 订单浏览查询
11	合同范本展示	会员可以浏览合同范本

20.2　MVC 实现原理

MVC（Model View Controller）即模型—视图—控制器。MVC 并不是 PHP 这类 B/S 结构程序中的概念，它本来是应用于 Desktop 程序中的一种软件设计模式。M 是指数据模型，V 是指用户界面，C 则是控制器。

在 B/S 结构的程序中，引入 MVC 的模式主要目的是将 M 层和 V 层实现代码分离。使用 MVC 模式的优点如下：

- 使同一个程序可以使用不同的表现形式；
- 应用 MVC 的模式可以使得增加一个应用程序变得规范化，这样团队之间就可以通过文档的形式进行项目协作，而不用关心具体的代码的实现。这也是众多开源的程序框架使用 MVC 模式的重要原因。

本节将详细讲解在 MVC 模式中各组成部分的具体作用。

20.2.1　MVC 基础

MVC 是一个设计模式，它强制性地使应用程序的输入、处理和输出分开。使用 MVC 应用程序被分成 3 个核心部件，即模型、视图、控制器。它们各自处理自己的任务。

- Model：模型层，负责完成具体的业务模型，多用于结合数据库完成具体的操作。
- View：视图层，页面显示模板，通常由模型层调用展示出最终效果。
- Controller：控制器，控制器接受用户的输入并调用模型和视图去完成用户的需求，同时也是程序的入口。

一个标准的 MVC 设计模式各个组成部分的流程如图 20.4 所示。在不同的应用中可能

会选择加强某一个核心部件，或者将两个部分融合在一起，比如，在某些情况下模型层（Model）和视图（View）会结合在一起。下面将对 MVC 模式中的各个组成部分的作用做具体的说明。

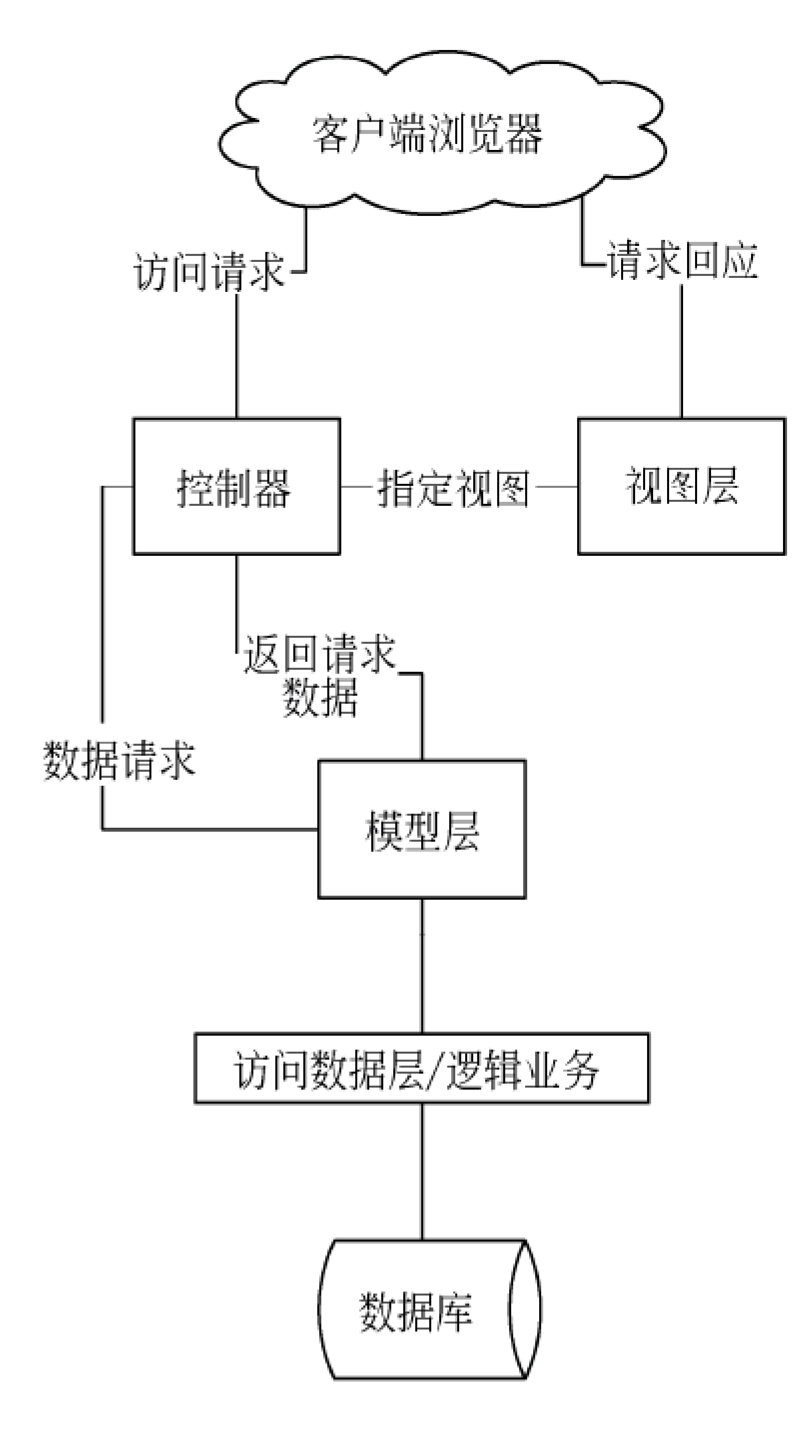


图 20.4　MVC 流程示意图

20.2.2　Model（模型）

模型表示企业数据和业务规则。在 MVC 的 3 个部件中，模型拥有最多的处理任务。例如，通常的网站应用就是通过创建对象，来调用数据库中的“信息”完成业务规则。理想情况下，模型返回的数据是中立的，就是说模型与数据格式无关，这样一个模型能为多个视图提供数据支持。由于应用于模型的代码只需写一次就可以被多个视图重用，所以极大地提高了开发的效率。

注意： 模型层中提到的业务模型是广义上的“业务”模型。在通常的应用中，模型层会被简单地描述成对数据“读/写”操作，这是不准确的，读者在实际应用过程中要注意体会。

下面以常见的对数据库的操作为例，方便读者对模型层有个直观地认识，方便后面的学习。示例代码如下：

```
if(!class exists("model")) {
    uses("model");                  //加载 model 基类
}
class newsinfo extends model{
    //获得当前索引 id
    function getIndexCode(){
        $db     = & $this->db();    //实例化数据库类
        $query  = "select tp newsinfo seq.nextval from dual";
        $rt     = $db->Query($query);
        $arr    = $db->getrow();
        $code   = $arr[0];
        //异常逻辑处理
        if(!empty($code)){
            Return $code;
        }else{
            Return false;
        }
    }
}
```

【代码解读】

以上代码通过加载 Model 的基类，获得初始化数据链接，然后定义一个 getIndexCode() 方法用来获得当前索引的 id。

20.2.3　View（视图）

视图是程序与用户交互的界面。对通常的 Web 应用程序来说，视图就负责展示层所有元素的集合，例如 Macromedia Flash 和 XHTML、XML/XSL、WML 等一些标识语言。在 Web 开发的过程中，一个很重要的工作就是如何显示越来越复杂的前端界面。在引入 MVC 的模式后，通过 MVC 模式组织代码，可以使应用程序处理根据不同的情况加载不同的视图。

注意：在视图层中，其实没有真正的处理发生，作为视图来说，只是作为一种输出数据并允许用户操纵的方式。

下面一段代码描述了一个典型的视图层的构成，这里使用开源的 Smarty 模板引擎作为页面模板的载体。经常使用 PHP 做 Web 的读者对它一定不陌生，这里通过示例做简单的回顾。

（1）定义一个模型类，它将完成模板引擎的配置和初始化的工作。模型类的核心代码如下：

```
class view{
    //模板方法
    function &tpl()
    {
        loadPlugIn("smarty/Smarty.class");          //加载模板引擎插件
        $tpl                    = & new Smarty();   //实例化模板引擎类
        $tpl->template_dir      = TEMPLATE_DIR;     //定义模板目录
        $tpl->compile_dir       = COMPILE_DIR;      //编译目录
        $tpl->cache_dir         = CACHE_DIR;        //文件缓存目录
        $tpl->left_delimiter    = '<%';             //定义模板标记 left
        $tpl->right_delimiter   = '%>';             //定义模板标记 right
        $tpl->caching           = 2;
        $tpl->compile_check     = true;
        $tpl->force_compile     = true;
        $tpl->cache_lifetime    = 60;               //设置缓存时间
        $tpl->debugging         = false;            //是否开发报错
        $tpl->assign("weburl",URL_DIR);
        $tpl->assign("image_dir",IMAGE_DIR);
        $tpl->assign("css_dir",CSS_DIR);
        $tpl->assign("js_dir",JS_DIR);
        $tpl->assign("webajax",WEBAJAX);
        return $tpl;
    }
}
```

（2）定义一个加载模板配置的方法，当控制器（Controller）选择对应的视图后，就可以正确地输出模板中的内容了。实现代码如下：

```
//加载模板引擎
function &loadView()
{
    //使用 uses()方法加载视图配置并初始化
    uses("view");                       //实例化视图基类
    $this->tpl = & view :: tpl();       //调用视图中的显示方法
    return $this->tpl;
}
```

20.2.4　Controller（控制器）

控制器是 MVC 架构中的核心，在一个应用中控制器会先接收用户的输入解析并调用对应的业务模型，然后加载指定的视图程序去选择对应的模板文件输出到用户的浏览器。所以当单击 Web 页面中的超链接和发送 HTML 表单时，控制器本身不输出任何东西，也不做任何处理。它只是接收请求并决定调用哪个模型构件去处理请求，然后再确定用哪个视图来显示模型处理返回的数据。

现在介绍控制器的具体代码设计。下面的代码是一个典型的控制器基类，读者通过这个例子可以了解到控制器的执行流程。示例代码如下：

```
<?
//控制器（controller）基类文件
if(!class exists('base')) {
    uses('base');
}
class controller extends base {
    var $module;                          //定义业务模型
    var $tpl;                             //定义模板
    //加载模板引擎
    function &loadView()
    {
        uses("view");                     //实例化视图基类
        $this->tpl = & view :: tpl();     //调用视图中的显示方法
        return $this->tpl;
    }

    /**
    * 函数名称： 系统默认的初始化操作，在业务逻辑处理之前运行；
    * 输入参数： void
    * 返回参数： void
    **/
    function Init() {
        if(WEBDEBUG) {
            $this->gpdebug();
        }
    }

    /**
    * 函数说明： 调试显示提交信息
    * 输入参数： void
    * 返回参数： string
    **/
    function gpdebug() {
        ob_start();                       //打开缓冲区
        var_dump(GP());
        $gpstr  = ob_get_contents();      //获得抛出的调试信息
        ob_end_clean();                   //关闭缓冲区
        $this-> debug("提交信息：",$gpstr);
    }

    /**
    * 函数名称： 系统默认的卸载操作，在业务逻辑处理完成之后运行；
```

```
    * 输入参数： void
    * 返回参数： void
    **/
    function onUnload() {
        //卸载相关操作
    }
}
?>
```

【代码解读】

上面程序的控制器基类文件中定义了 4 个基本的方法，其含义分别是：

（1）加载模板引擎的 loadView()方法。

（2）系统默认的初始化操作的 Init()方法。

（3）调式显示提交信息 gpdebug()方法。

（4）系统默认的卸载操作 onUnload()方法。

20.3　数据库设计

图片交易平台的数据库设计是为了满足访问者的业务需求。本节将进行数据的抽象、建模，确定数据库的结构、数据存储过程及触发器的设计，使解决方案中的需求得到满足。它是开发过程中的重要参考，同时也是开发过程能够健康有序进行的基础。数据库设计是项目开发过程中最重要的部分，可以说数据库的功能设计体现了用户的基本需求。

20.3.1　需求分析

图片交易平台是艺术品数据库的一个应用平台、是对艺术品数据库中数据的一个具体的操作平台。图片交易平台的数据来源分两个部分：

（1）可写数据，由交易平台自己录入的产生（如资讯、会员订单、会员相关数据）。

（2）只读数据，从艺术品数据库，或者通过其他合作网站同步过来（如艺术品资料、艺术家资料、艺术品分类等艺术品相关数据）。

图片交易平台的数据建设也分两个阶段（或者是数据同步分成两个方面），在艺术品数据库升级期间，只读数据部分从合作网站数据库同步；当艺术品数据库升级完成后，只读数据从艺术品数据库同步。

注意：在同步合作网站或者艺术品数据库的数据库时，可以不同步数据结构、只同步所需数据。

图片交易平台的数据同步流程如图 20.5 所示，该图很好地说明了图片交易平台的数据同步过程。

20.3.2　数据库 E-R 关系图

E-R 图，即实体-联系图（Entity Relationship Diagram），是指提供表示实体型、属性和联系的方法，用来描述现实世界的概念模型，在数据库设计中使用这种模式进行数据描

述和设计。图片交易平台数据库 E-R 关系图如图 20.6 所示。

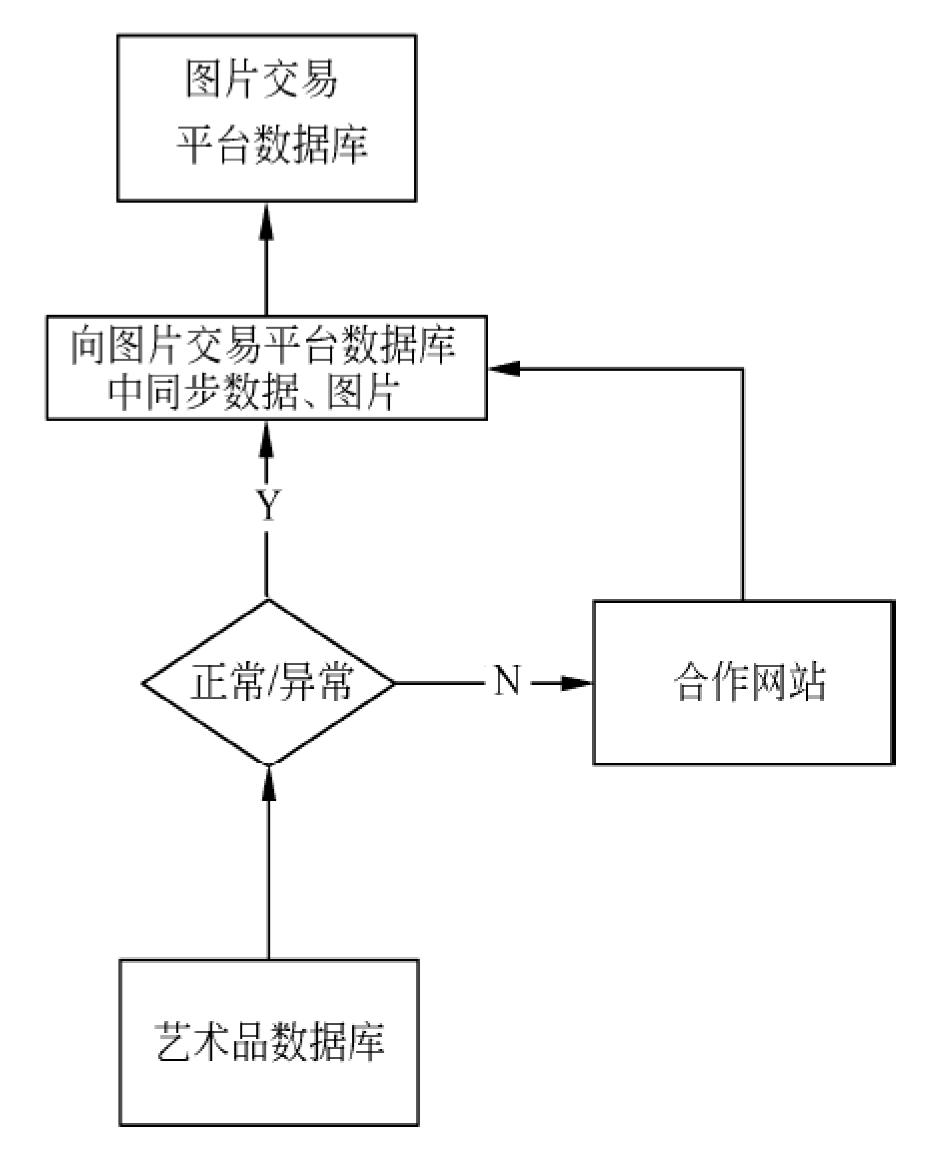


图 20.5　图片交易平台数据同步流程图

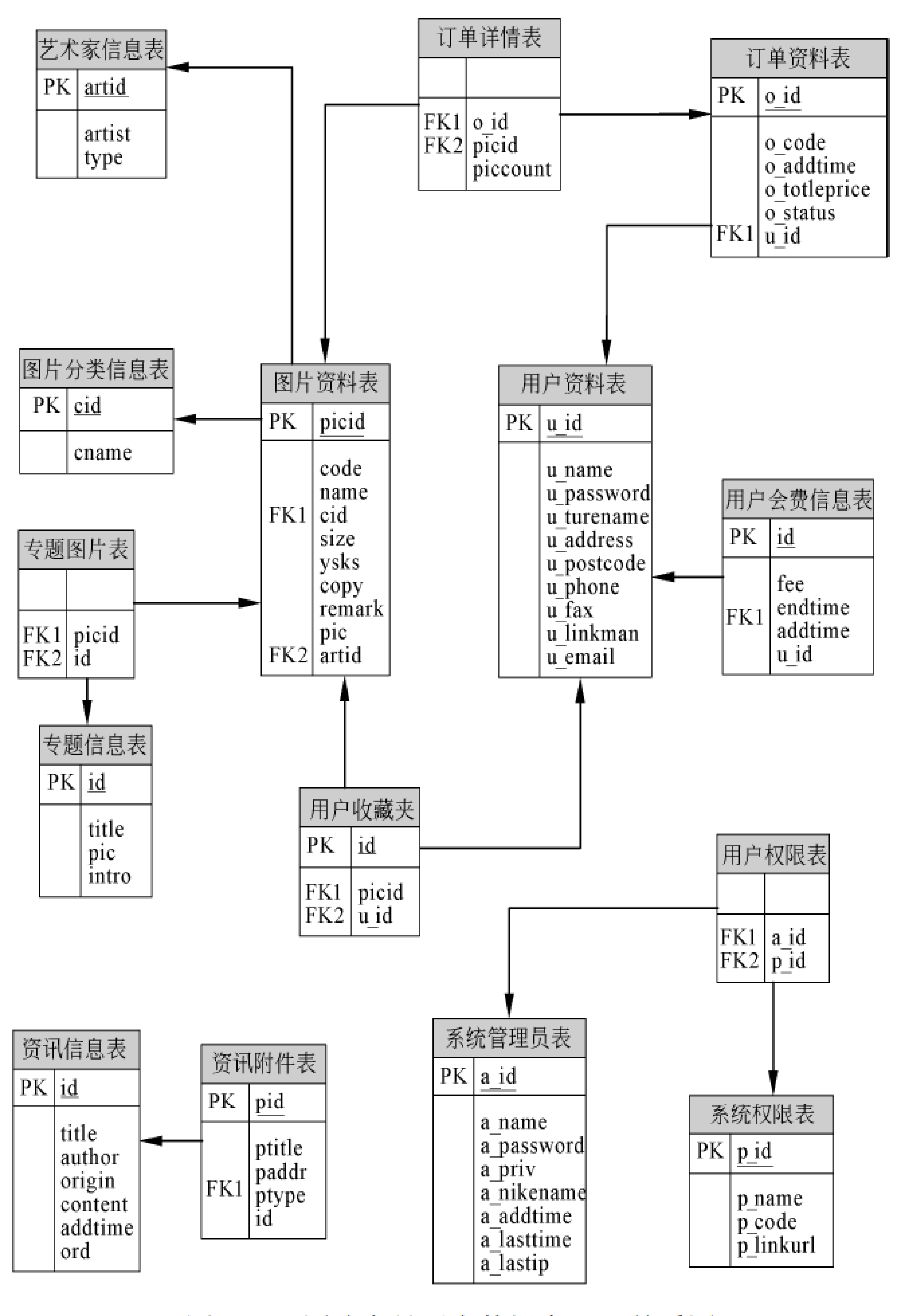


图 20.6　图片交易平台数据库 E-R 关系图

20.3.3　数据表设计

图片交易平台一共是 15 张表，按照功能分为 3 类：

（1）图片信息相关表（如图片分类信息表、图片资料表）。

（2）后台管理相关表。

（3）业务相关表（如订单详情表和订单资料表）。

下面通过表格的形式，具体介绍表的数据结构和字段含义。

（1）管理用户表（tp_Admin）：存储管理员资料。见表 20.2。

表 20.2　管理用户表（tp_Admin）

字　段	名　称	类　型	NULL	备　注
A_id	管理员序号	Integer	No	主键 索引
A_name	管理员用户名	Char(20)	No	
A_password	管理员密码	Char(20)	No	
A_priv	管理员权限	Char(10)	No	
A_nickname	管理员昵称	Char(20)	No	
A_addtime	管理员创建时间	datetime	No	
A_lasttime	最后登录时间	datetime	No	
A_lastip	最后登录 ip	Char(20)	No	

（2）系统权限表（tp_priv）：保存系统权限。见表 20.3。

表 20.3　系统权限表（tp_priv）

字　段	名　称	类　型	NULL	备　注
P_id	权限序号	Integer	No	主键 索引
P_name	权限名称	Char(20)	No	
P_code	权限编号	Char(20)	No	
P_linkurl	权限对应页面	Char(10)	No	

（3）管理员权限表（tp_admpriv）：保存管理员对应的权限。见表 20.4。

表 20.4　管理员权限表（tp_admpriv）

字　段	名　称	类　型	NULL	备　注
P_id	权限序号	Integer	No	外键（关联 tp_priv.p_id）
A_id	管理员序号	Integer	No	外键（关联 tp_admin.a_id）

（4）会员资料表（tp_users）：保存会员资料。见表 20.5。

表 20.5　会员资料表（tp_users）

字　段	名　称	类　型	NULL	备　注
U_id	序号	Integer	No	主键 索引
U_name	会员用户名	Char(20)	No	
U_password	会员密码	Char(20)	No	
U_truename	机构名称	Char(60)	No	

续表

字　段	名　称	类　型	NULL	备　注
U_address	联系地址	Char(100)	No	
U_postcode	邮编	Char(10)	No	
U_phone	电话	Char(20)	No	
U_fax	传真	Char(20)	No	
U_linkman	联系人	Char(20)	No	
U_email	email	Char(80)	No	

（5）用户会费资料表（tp_userfee）：保存用户会费资料。见表 20.6。

表 20.6　用户会费资料表（tp_userfee）

字　段	名　称	类　型	NULL	备　注
id	序号	Integer	No	主键 索引
U_id	用户 id	Integer	No	外键（关联 tp_users.u_id）
fee	费用金额	Char(20)	No	
endtime	到期时间	Datetime	No	
addtime	录入时间	Datetime	No	

（6）艺术图片资料表（tp_picinfo）：艺术品资料详情（名称、作者等）。见表 20.7。

表 20.7　艺术图片资料表（tp_picinfo）

字　段	名　称	类　型	NULL	备　注
Picid	图片序号	Integer	No	主键 索引
Code	图片编号	Char(20)	No	
Name	图片名称	Char(20)	No	
cid	图片分类	Char(10)	No	外键（关联）
Size	图片尺寸	Char(20)	No	
Ysks	印刷开数	Char(20)	No	
Copy	版权	Char(40)	No	
Remark	备注	Text	No	
Pic	图片路径	Char(128)	No	
artid	作者（艺术家 id）	Char(20)	No	外键（关联）

（7）订单信息表（tp_orderinfo）：记录订单信息（用户 id、订单状态）。见表 20.8。

表 20.8　订单信息表（tp_orderinfo）

字　段	名　称	类　型	NULL	备　注
O_id	订单序号	Integer	No	主键 索引
O_code	订单编号	Char(20)	No	
U_id	用户 id	Integer	No	外键（关联 tp_users.u_id）
O_addtime	下单时间	Datetime	No	
O_totalprice	订单总价格	Char(20)	No	
O_status	订单状态	Integer	No	值的范围（0 或 1）0 表示未完成状态，1 表示已完成状态

（8）订单详情信息表（tp_ordergoods）：记录物品详情（图片 id、个数等）。见表 20.9。

表 20.9　订单详情信息表（tp_ordergoods）

字　段	名　称	类　型	NULL	备　注
O_id	订单 id	Integer	No	外键（关联 tp_orderinfo.o_id）
picid	图片 id	Integer	No	外键（关联 tp_picinfo.picid）
Piccount	图片个数	Integer	No	

（9）艺术家表（tp_artist）：记录艺术家相关信息。见表 20.10。

表 20.10　艺术家表（tp_artist）

字　段	名　称	类　型	NULL	备　注
artid	艺术家编号	Char(20)	No	主键 索引
artist	艺术家用户名	Char(64)	No	
type	艺术家名称类别	Char(20)	No	

（10）艺术图片分类表（tp_category）：记录图片分类详情。见表 20.11。

表 20.11　艺术图片分类表（tp_category）

字　段	名　称	类　型	NULL	备　注
cid	分类编号	Char(15)	No	主键 索引
Cname	分类名称	Char(64)	No	

（11）专题信息表（tp_topicinfo）：记录专题信息。见表 20.12。

表 20.12　专题信息表（tp_topicinfo）

字　段	名　称	类　型	NULL	备　注
id	专题序号	Integer	No	主键 索引
Title	专题标题	Char(20)	No	
Pic	专题标题图片	Char(20)	No	
intro	专题介绍	Char(10)	No	

（12）专题图片表（tp_topicpic）：记录专题中图片 id。见表 20.13。

表 20.13　专题图片表（tp_topicpic）

字　段	名　称	类　型	NULL	备　注
id	专题序号	Integer	No	外键（关联 tp_topicinfo.id）
picid	图片 id	Char(20)	No	外键（关联 tp_picinfo.picid）

（13）用户收藏夹（tp_collection）：记录用户收藏夹中的信息。见表 20.14。

表 20.14　用户收藏夹（tp_collection）

字　段	名　称	类　型	NULL	备　注
id	收藏夹序号	Integer	No	主键 索引
Uid	会员 id	Integer	No	外键（关联 tp_users.u_id）
Picid	图片 id	Char(20)	No	外键（关联 tp_picinfo.picid）

（14）资讯信息表（tp_newsinfo）：保存资讯信息内容。见表 20.15。

表 20.15　资讯信息表（tp_newsinfo）

字　段	名　称	类　型	NULL	备　注
id	序号	Integer	No	主键 索引
Title	标题	Char(128)	No	
Author	作者	Char(20)	No	
Origin	来源	Char(40)	No	
Content	资讯内容	Text	No	
addtime	创建时间	datetime	No	
Ord	排序	Integer	No	

（15）资讯图片表（tp_newspic）：资讯对应的图片详情。见表 20.16。

表 22.16　资讯图片表（tp_newspic）

字　段	名　称	类　型	NULL	备　注
Pid	资讯图片序号	Integer	No	主键 索引
Ptitle	图片说明	Char(64)	No	
Paddr	图片地址	Char(128)	No	
Ptype	图片类型	integer	No	取值范围（0～3）0 为图片、1 为 flash、2 为视频、3 为其他
id	新闻序号	Integer	No	外键（关联 tp_newsinfo.id）

20.4　图片搜索及展示

图片交易平台中，最常使用的功能就是图片的展示功能。为了便于使用者更快捷地找到需要的图片，还需要在系统中加入图片关键字的搜索功能，通过这些功能用户就可以直接找到需要的图片。本节将为读者介绍图片的展示功能、搜索功能的设计过程及具体的实现步骤，也是通过 MVC 的方式完成一个具体的流程模块。读者在学习的过程中，应着重注意程序是如何通过 MVC 的模式来完成具体的功能，以及各个“部件”是如何协作的。

20.4.1　图片列表

图片交易平台是一个交易的平台系统，读者可以把它想象成一个完成图片交易市场。任何一个用户进入市场都会“看”到各种“商品”，然后通过对比、辨别、筛选，才会确定想要买的“商品”。这个“看”的过程具体到图片交易平台就是“图片列表”功能。图片列表功能是通过读取数据库中全部的图片信息，然后将图片以列表的方式展示给用户。

通过前面的学习，相信读者已经可以通过一般的代码组织模式，独立地完成该功能。现在介绍在 MVC 的模式下是如何实现图片列表功能的。

1．控制器

正如前面介绍过的“控制器”是 MVC 的核心，它负责“组织”其他“部件”完成指

定的功能。根据 MVC 定义的规则，创建图片显示控制器（c_artimage.php），定义控制器的访问样式如下：

```
http://localhost/pt/index.php?/c/artimage/
```

首先创建构造函数用来初始化图像显示控制器，然后定义控制器的默认“动作”。图片控制器代码框架代码如下：

```
//是否正常读取
if(!defined('APPNAME')) {
    exit("没有权限");
}
class c artimage extends controller {
//构造函数
    function c artimage() {
        loadModel("artimage");          //加载图片模型类
        $this->ai = & new artimage;     //实例化图片模型类
    }
//默认动作
    function onMain() {
        if(GP('act')=="") {
            $this -> onlist();          //图片列表展示方法
        }
    }
}
```

【代码解读】

function c_artimage()构造函数是类中的一个特殊函数，当使用 new 操作符创建一个类的实例时，构造函数将会自动调用。当函数与类同名时，这个函数将成为构造函数。如果一个类没有构造函数，则自动调用基类的构造函数。

构造函数通常是用来初始化类的基础设置的，这里使用 loadModel 方法加载“artimage”模型。该模型被认为是处理图片显示相关功能，这些功能通常是对应具体的“业务”的，因此也被称为“业务模型”。

function onMain()函数是控制器默认加载动作，这里具体的就是加载 onlist()这个方法，它的功能被认为是显示全部图片列表。

下面的代码是 onlist()的具体实现：

```
function onlist() {
    uses("pagebar");                            //加载分页基类
    $ai     = &$this->ai;
    $pb     = new pagebar;
    /*分页功能参数*/
    $pz     = 12;                               //分页跨度
    $lp     = url('artimage','list',array());
    $tr     = $ai->getArtimageCount();          //获得图片数目
    /*执行分页功能*/
    $pbar   = $pb-> execute($tr,$pz,$lp);
    $pbar['startrow']   = 0;                    //起始记录
    $pbar['endrow']=10;                         //跨度
    $list   = $ai->getArtimageList(0,$pbar['startrow'],$pbar['endrow']);
                                                //获得图片列表
    $clist  = $ai->getCategoryList();           //获得分类列表
```

```
    $tpl    = & $this -> loadview();             //加载视图文件
    $tpl -> assign("pagebar",$pbar);
    $tpl -> assign("list",$list);
    $tpl -> assign("clist",$clist);
    $tpl -> display('list.html');                //加载视图列表页面
}
```

【代码解读】

Onlist()方法首先加载 pagebar 这个基础类库，该类库是用来处理图片翻页相关问题的。分页功能代码如下：

```
/*执行分页功能*/
    $pbar   = $pb->_execute($tr,$pz,$lp);
```

实例化该类后，将以数组的方式返回图片处理相关的内容。

在构造函数的时候初始化了图片显示模型类，调用其中的$ai->getArtimageCount()方法获得当前图片的总数，然后通过$ai->getArtimageList()方法获得图片列表数组，再加载对应视图文件（list.html）。

2．模型

在控制器部分规定了具体模型的功能，这些“业务”功能共同完成这个显示图片列表的业务流程。

下面具体介绍实现控制器中涉及的“业务”功能。先获取全部图片列表的函数 function getArtimageList($ord,$startnum,$endnum)。实现获取全部图片列表功能的代码如下：

```
/**
* 函数名称： 获得全部交易图片的列表
* 输入参数：
    (string)$ord:   排序
    (int)$startnum: 读取记录的开始点
    (int)$endnum:   读取记录的结束点
* 返回参数：
    (array):操作成功返回记录数组，反之，返回 false
**/
function getArtimageList($ord,$startnum,$endnum){
    $db     = & $this->db("oracle_page");         //实例化数据库连接
    $ordstr = $this->ordToStr($ord);
    $query  = "SELECT * FROM tp_picinfo $ordstr";  //执行 SQL
    $rt     = $db->PageQuery($query,$startnum,$endnum);
    $rtarr  = $db->getrows($rt);                  //获得图片列表数组
    if($rtarr) {
        Return $rtarr;                            //返回列表数组
    }else {
        Return false;
    }
}
```

【代码解读】

该函数的核心是将数据从 tp_picinfo 这张表中全部取出，其中引入了实现分页相关的两个设置，即$startnum 参数和$endnum 参数。

获得全部交易图片的总记录数 function getArtimagecount()的代码如下：

```
/**
* 函数名称：  获得全部交易图片的总记录数
* 输入参数：  void
* 返回参数：  int
**/
function getArtimagecount(){
    $db     = & $this->db("oracle page");
    $query  = "SELECT count(*) as num FROM tp_picinfo";
    $rt     = $db->Query($query);            //执行 SQL
    $rtarr  = $db->getrow($rt);              //获得记录总数
    if($rtarr) {
        Return $rtarr[0];                    //返回记录总数
    }else {
        Return false;
    }
}
```

【代码解读】

该函数的核心是计算 tp_picinfo 这张表行数并将结果赋值给 num。获得全部图片类别列表函数 function getCategoryList()代码如下：

```
    /**
    * 函数名称：  获得全部图片类别的列表
    * 输入参数：  void
    * 返回参数：
        (array):操作成功返回记录数组，反之，返回 false
    **/
    function getCategoryList(){
        $db     = & $this->db("oracle page");      //实例化数据库连接
        $query  = "SELECT * FROM tp_category";
        $rt     = $db->Query($query);              //执行 SQL
        $rtarr  = $db->getrows($rt);               //获得图片类别记录集
        if($rtarr) {
            Return $rtarr;                         //返回图片类别数组
        }else {
            Return false;
        }
    }
}
```

【代码解读】

该函数的核心将数据从 tp_category 这张表中全部取出，然后通过 getrows()方法将数据值数组取出。

3. 视图

在控制器中设定了加载 list.html，该文件是显示图片列表的视图。该文件的内容显示部分使用模板引擎定义的标签，将定义好的图片列表数组循环的方式显示出来。该文件中显示图片列表的部分的代码如下：

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
    <%section name=item loop=$list%>
    <%if $smarty.section.item.index % 4 ==0 %>
    <tr>
```

```
        <%/if%>
        <td width="25%">
        <table width="27%" border="0" align="center" cellpadding="0"
            cellspacing="0">
            <tr>
                <td><img id="imglist<%$smarty.section.item.index%>"
                src="<%$image dir%><%$list[item].Code%>.gif" width="155"
                height="100" /></td>
            </tr>
            <tr>
                <td height="30">
                <table width="100%" border="0" cellpadding="0" cellspacing="0">
                    <tr>
                        <td width="87%" class="zt"><a href="?/c/artimage/
                        act/show/id/<%$list[item].PICID%>"><font color=
                        "#0066FF"><%$list[item].Name|truncate:10:""%>
                         </font></a></td>
                        <td width="13%"><img src="<%$image dir%>fa.gif"
                        width="14" height="14" /></td>
                    </tr>
                </table>
                </td>
            </tr>
            <tr>
                <td class="zt"><%$list[item].Code|truncate:10:""%></td>
            </tr>
            <tr>
                <td class="zt"><a href="?/c/artimage/act/like/cid/<%$list
                [item].cid%>">类似图片</a>
                <a href="?/c/collection/act/add/id/<%$list[item].Picid%>
                ">收藏</a></td>
            </tr>
        </table>
        </td>
        <%if $smarty.section.item.index % 4 ==3 || $smarty.section. item.
        last%>
    </tr>
    <tr>
        <td> </td>
        <td> </td>
        <td> </td>
        <td> </td>
    </tr>
    <%/if%>
<%/section%>
    <tr>
        <td colspan="4">
        <table width="100%" border="0" cellpadding="0" cellspacing="0">
            <tr>
                <td width="278" align="right"><a href="<%$pagebar.
                linkparam%>page/<%$pagebar.prevpage%>"><img
                    src="<%$image_dir%>left.gif" width="18" height="18"
                border="0" /></a></td>
                <td width="7" align="right"> </td>
                <td width="141" class="zt"><%section name=pglist
                loop=$pagebar.stride array%> <span class="nu"><a href="
                <%$pagebar.linkparam%>page/<%$pagebar.stride array[pglist]
                %>"><%$pagebar.stride array[pglist]%></a></span>
                <%if not $smarty.section.pglist.last%>.<%/if%>
                <%/section%></td>
                <td width="10"> </td>
```

```
                <td width="273"><a href="<%$pagebar.linkparam%>page/
                <%$pagebar.nextpage%>"><img
                    src="<%$image dir%>right.gif" width="18" height="18"
                border="0" /></a></td>
            </tr>
        </table>
        </td>
    </tr>
    <tr>
        <td> </td>
        <td> </td>
        <td> </td>
        <td> </td>
    </tr>
</table>
```

【代码解读】

上述模板代码中负责循环部分的模板引擎标签样式如下：

```
<%section name=item loop=$list%>
    <%if $smarty.section.item.index % 4 ==0 %>
    <tr>
    <%/if%>
<%/section%>
```

这里需要注意的是，模板引擎定义了显示字节数的方法，代码如下：

```
<%$list[item].Code|truncate:10:""%>
```

Truncate:X 中 X 的值就是需要显示的字节数。

在浏览器中执行下面的链接：

```
http://localhost/pt/index.php?/c/artimage/
```

执行该链接后将显示图片列表页面，效果如图 20.7 所示。

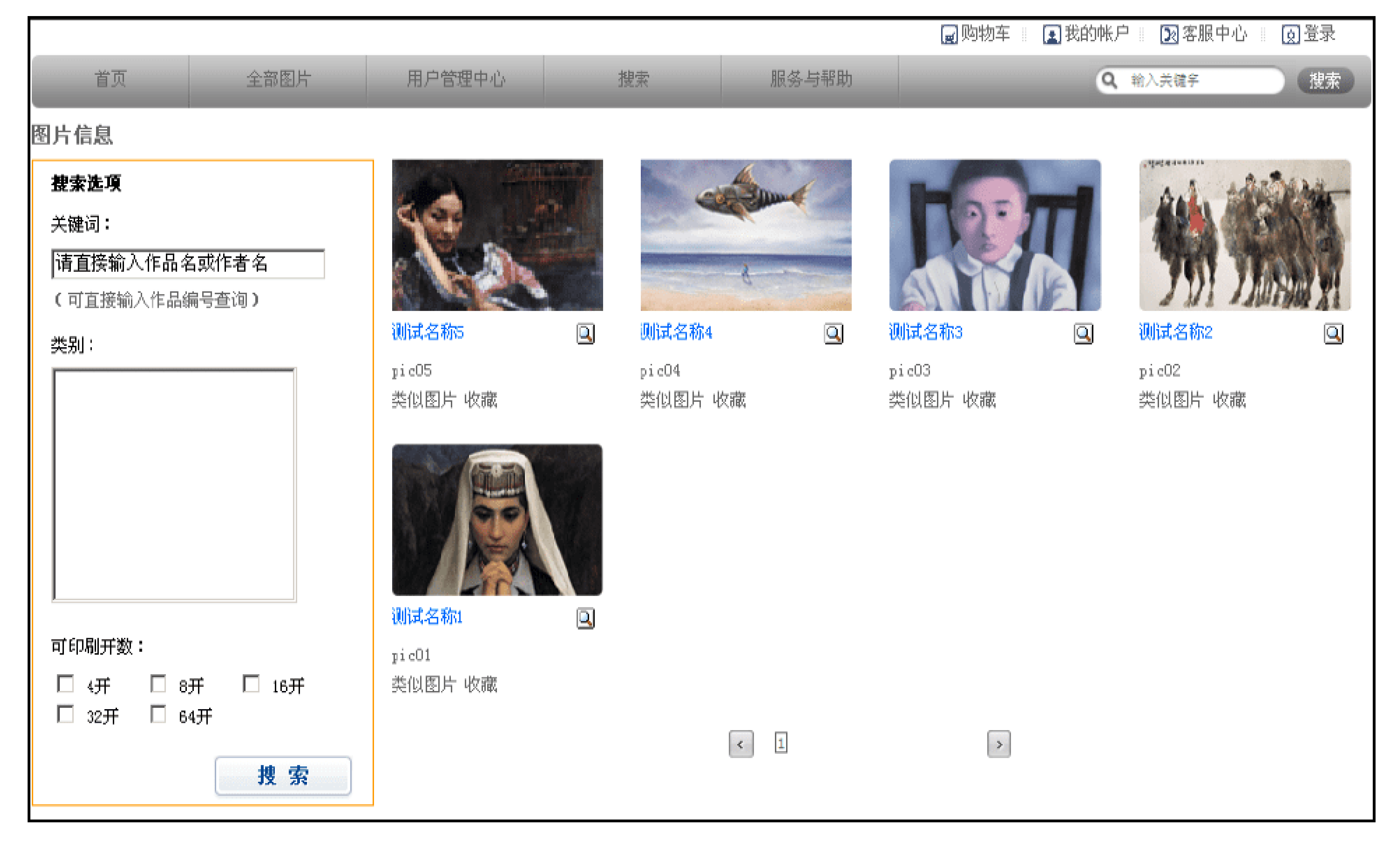


图 20.7　图片列表

20.4.2　图片搜索

图片搜索是用户在进行图片浏览时的辅助功能，这个功能是为那些有“目标”的用户设计的。用户可以通过图片名称关键字来检索对应图片，当有多个符合条件的结果时，图片将以列表的形式展示。

1．控制器

创建控制器文件 c_search.php，下面介绍控制器部分的设计。图片搜索的控制器代码如下：

```
//是否正常读取
if(!defined('APPNAME')) {                         //权限检查
    exit("没有权限");
}
class c search extends controller {
    function c search() {
        loadModel("artimage");                    //加载图片业务模型类
        $this->ai = & new artimage;               //实例化图片业务模型
    }
//默认加载动作
    function onMain() {
        if(GP('act')=="") {
            $this -> onshow();                    //默认加载 onshow()方法
        }
    }
//图片检索动作
    function onshow() {
        $ai     = &$this->ai;
        $clist  = $ai->getCategoryList();         //获得分类列表
        $tpl    = & $this -> loadview();          //实例化视图模型
        $tpl -> assign("clist",$clist);           //生成分类列表标签
        $tpl -> display('search.html');           //加载搜索页面文件
    }

    function onsear() {
        $this->search(GP('keyword'));             //获得关键字方法
    }
//根据关键字搜索
    function onsearch() {
        $this->search(GP('keywords'),GP('cat'),GP('ysks'));
    }
//根据传入关键字获得图片检索结果，并以数组形式返回
    function search($keyword,$cat='',$ysks='') {
        uses("pagebar");                          //加载分页类
        $ai     = &$this->ai;
        $pb     = new pagebar;                    //实例化分页类
        $pz     = 12;
        $lp     = url('search','search',array('keywords'=>$keyword,
                  'cat'=>$cat,'ysks'=>$ysks));
        $tr     = $ai->searchArtimageCount($keyword,$cat,$ysks);
                                                  //获取该次搜索图片数量
        $pbar   = $pb->_execute($tr,$pz,$lp);
```

```
        $pbar['startrow']  = 0;              //起始记录
        $pbar['endrow']=10;                  //跨度
        $list   = $ai->searchArtimage($keyword,$cat,$ysks,0,$pbar
                  ['startrow'],$pbar['endrow']);
        $clist  = $ai->getCategoryList();    //获得分类列表
        $tpl    = & $this -> loadview();
        $tpl -> assign("pagebar",$pbar);
        $tpl -> assign("list",$list);
        $tpl -> assign("clist",$clist);
        $tpl -> display('list.html');        //加载列表页面文件
    }
}
```

【代码解读】

图片检索控制器首先初始化控制器的设置并加载默认的方法，该方法被定义为显示搜索引导页面。代码如下：

```
//构造函数
    function c search() {
        loadModel("artimage");               //加载图片业务模型类
        $this->ai = & new artimage;          //实例化图片业务模型
    }
//默认动作
    function onMain() {
        if(GP('act')=="") {
            $this -> onshow();
        }
    }
```

图片检索控制的构造函数也和图片列表部分一样，是加载 artimage 图片商品类。该类被定义为处理图片相关的“业务”动作。加载该模型的代码如下：

```
loadModel("artimage");
```

搜索动作函数是控制器的核心方法（function search($keyword,$cat='',$ysks='')，该方法被定义为根据传入的参数值返回对应的图片列表信息。

2．模型

图片检索部分的模型就是图片商品类（artimage），该类中的部分方法被设计用来处理图片检索及结果返回。下面来介绍与图片检索有关的业务模型。

搜索交易图片功能模型函数 function searchArtimage()，代码如下：

```
/**
* 函数名称： 搜索交易图片
* 输入参数：
    (string)$keyword:   检索关键字
    (string)$ysks:      印刷开数
    (string)$cat:       类别名称
    (int)$ord:          排序编号
    (int)startnum:      读取记录的开始点
    (int)endnum:        读取记录的结束点
* 返回参数： void
**/
function searchArtimage($keyword,$cat,$ysks,$ord,$startnum,$endnum) {
```

```
    if(!empty($ord)) {                              //检查排序编号是否存在
        $ordstr= $this->ordToStr($ord);
    }
    $db     = & $this->DB("oracle_page");           //实例化数据库
    $query  = $this->searchsql($keyword,$cat,$ysks,0);
    $query  = $query.$ordstr;
    $rt     = $db->pageQuery($query,$startnum,$endnum);
    $rtarr  = $db->getrows($rt);                    //获得搜索结果记录集
    if($rtarr) {
        Return $rtarr;                              //返回搜索结果记录集
    }else {
        Return false;
    }
}
```

【代码解读】

搜索交易图片函数中，并没有直接操作数据完成图片检索动作，而是将复用程度很高的合成对比功能抛给 searchsql()函数。该函数被定义用来合成检索用的 SQL 语句并返回对应的结果。

获取图片记录数函数 function searchArtimageCount()，代码如下：

```
/**
* 函数说明： 执行 sql 获取艺术图片的记录数
* 输入参数：
    (string)$keyword:   检索关键字
    (string)$ysks:      印刷开数
    (string)$cat:       类别名称
* 返回参数： int
**/
function searchArtimageCount($keyword,$cat,$ysks) {
    $db     = & $this->DB("oracle_page");           //实例化数据库连接
    $query  = $this->searchsql($keyword,$cat,$ysks,1);
    $rt     = $db->Query($query);                   //执行 SQL
    $rtarr  = $db->getrow($rt);                     //获得图片记录数结果集
    if($rtarr) {
        Return $rtarr[0];                           //返回图片记录数结果集
    }else {
        Return false;
    }
}
```

合成搜索查询 SQL 函数 function searchsql()，代码如下：

```
/**
* 函数说明： 合成搜索查询 sql
* 输入参数：
    (string)$keyword:   检索关键字
    (string)$ysks:      印刷开数
    (string)$cat:       类别名称
    (int)$iscount:      是否是查询总记录数
* 返回参数： string
**/
function searchsql($keyword,$cat,$ysks,$iscount=0) {
    if(!empty($keyword)) { //判断关键字
        $conditions     .= "and (t1.Name like '%$keyword%')";
```

```
    }
    if(is_array($cat)) {        //判断类别名称
        $conditions    .= "and t1.cid in ('".explode("','",$cat)."')  ";
    }
    if(is array($ysks)) {       //判断印刷开数
        $conditions    .= "and t1.ysks in ('".explode("','",$ysks)."') ";
    }
    $conditions         = (empty($conditions)) ? "" : " where ".substr
    ($conditions,3);
    if($iscount==0) {
        $query  = "SELECT t1.* FROM tp_picinfo t1 $conditions order by t1.
        Picid desc";
    }else{
        $query  = "SELECT count(*) num $fields FROM tp_picinfo t1 $tables
        $conditions ";
    }
    Return $query;
}
```

【代码解读】

该函数通过判断传入的关键字的类型，合成对应的 SQL 语句，实现对不同类型关键字结果的匹配。

根据排序编号获得排序字串函数 function ordToStr($ord)，根据不同的排序方式合成 SQL 语句，代码如下：

```
/**
* 函数名称：  根据排序编号获得排序字串
* 输入参数：
    (int)$ord:  排序编号
* 返回参数：
    (string):   排序字符串
**/
function ordToStr($ord) {
    $ord    = intval($ord);
    switch($ord){
        case 1:     //图片 id 倒序
            $rt = " order by picid desc";
            break;
        case 2:     //图片 id 升序
            $rt = " order by picid asc";
            break;
        case 3:     //标号倒序
            $rt = " order by code desc";
            break;
        case 4:     //标号升序
            $rt = " order by code asc";
            break;
        default :
            $rt = " order by picid desc";
    }
    Return $rt;
}
```

获得全部图片类别的列表函数 getCategoryList()，该函数代码如下：

```
    /**
    * 函数名称：  获得全部图片类别的列表
    * 输入参数：  void
```

```
    * 返回参数：
        (array):操作成功返回记录数组，反之，返回 false
    **/
    function getCategoryList(){
        $db     = & $this->db("oracle page");
        $query  = "SELECT * FROM tp_category";
        $rt     = $db->Query($query);
        $rtarr  = $db->getrows($rt);
        if($rtarr) {
            Return $rtarr;
        }else {
            Return false;
        }
    }
}
```

【代码解读】

该函数通过遍历 tp_category 表获取分类表中的全部数据，并以数组的方式返回结果。读者在实际的使用中，需要注意判断返回结果的异常状态。

3．视图

在图片检索部分控制器中定义了两种加载模板，在默认情况下加载 search.html 文件，该文件是独立的检索引导界面。调用 search()方法会以列表的形式返回界面，这个视图文件之前已经介绍过了。

下面介绍 search.html 文件，该文件的功能是显示图片检索引导界面。核心部分代码如下：

```
    <table width="80%" border="0" align="center" cellpadding="0"
    cellspacing="0">
    <tr>
        <td align="right" class="zt">关键词：</td>
        <td class="zt">
        <input name="keywords" type="text" id="keywords" value="请直接输入
        作品名或作者名" size="50"
            onblur="if (this.value=='') this.value='请直接输入作品名或作者名';"
            onfocus="if (this.value=='请直接输入作品名或作者名') this.value=
            ''; " /></td>
    </tr>
    <tr>
        <td width="23%" align="right" class="zt"> </td>
        <td width="77%" class="zt">（可直接输入作品编号查询）</td>
    </tr>
    <tr>
        <td align="right" class="zt"> </td>
        <td class="zt"> </td>
    </tr>
    <tr>
        <td align="right" valign="top" class="zt">类别：<br />
        </td>
        <td class="zt">
        <select name="cat[]" size="10" multiple="multiple" style="width:
        180px;">
            <%section name=item1 loop=$clist%>
            <option value="<%$clist[item1].cid%>"><%$clist[item1]. cname%>
            <%/section%>
```

```
        </select></td>
    </tr>
    <tr>
        <td align="right" class="zt"> </td>
        <td class="zt"> </td>
    </tr>
    <tr>
        <td align="right" class="zt">可印刷开数：</td>
        <td class="zt"><span class="ti">
        <input type="checkbox" name="ysks[]" value="4" /> 4 开
        <input type="checkbox" name="ysks[]" value="8" /> 8 开
        <input type="checkbox" name="ysks[]" value="16" /> 16 开
        <input type="checkbox" name="ysks[]" value="32" /> 32 开
        <input type="checkbox" name="ysks[]" value="64" /> 64 开</span></td>
    </tr>
    <tr>
        <td class="zt"> </td>
        <td class="zt"> </td>
    </tr>
    <tr>
        <td class="zt"> </td>
        <td><input type="image" src="<%$image dir%>search 2.gif" width=
        "102"height="29" /></td>
    </tr>
</table>
```

【代码解读】

该模板文件的核心是定义了关键字和检索条件的表单。这里需要注意的是，复合条件的实现是通过不同的参数输出的方式实现的。

在浏览器中执行下面的链接：

```
http://localhost/pt/index.php?/c/search/
```

执行该链接后将出现图片检索页面，效果如图 20.8 所示。



图 20.8　图片检索

在搜索框中输入“测试名称 4”文件测试图片检索的结果，检索结果如图 20.9 所示。



图 20.9　图片检索结果页面

20.4.3　图片详情

用户通过对比、筛选找到对应的图片后，需要了解图片的详细信息，这时候用户可以通过图片详情模块获得相应信息，并且该模块还是“加入购物车”、“查看类似图片”和“立即订购”的入口。

下面具体介绍该部分的设计和代码实现。创建控制器文件 c_artimage.php，文件中加入 onshow()方法，用来实现图片详情功能。

1．控制器

在 artimage 控制器文件中加入 onshow()方法，该部分代码如下：

```
function onshow() {
    $ai      = &$this->ai;
    $id      = intval(GP('id'));
    $info    = $ai->getArtimageInfoById($id);    //获得指定 id 的图片详情
    $tpl     = & $this -> loadview();            //实例化视图
    $tpl -> assign("info",$info);
    $tpl -> display('pic_list.html');            //加载视图模板
}
```

【代码解读】

该方法通过 GP 函数安全获取图片的 id 编号，此后通过$ai->getArtimageInfoById($id);方法获得图片详情，然后加载对应的图片详情模板。

由于图片详情部分涉及大尺寸图片和对应的小尺寸图片的显示问题，所以创建 3 个方法来处理。代码如下：

```
//相似图片的大图
```

```
    function onlike() {
        $this->xspic(1);
    }
//相似图片的小图
    function onlikeframe() {
        $this->xspic(2);
    }
//处理不同图片尺寸的显示
    function xspic($xs=1) {
        uses("pagebar");
        if($xs==1) {
            $pz     = 12;
            $tplym  = 'list.html';              //正常图片列表页面
            $act    = 'like';
        }else {
            $pz     = 5;
            $tplym  = 'xspiclist.html';         //相似图片列表页面
            $act    = 'likeframe';
        }
        $cid    = intval(GP('cid'));            //获得分类 id
        $ai     = &$this->ai;
        $pb     = new pagebar;                  //实例化分页类
        $lp     = url('artimage',$act,array('cid'=>$cid));//生成图片链接
        $tr     = $ai->searchArtimageCount($keyword,$cat,$ysks);
        $pbar   = $pb-> execute($tr,$pz,$lp);
        $list   = $ai->searchArtimage($keyword,$cat,$ysks,0,$pbar
        ['startrow'],$pbar['endrow']);
        $clist  = $ai->getCategoryList();       //生成分类列表
        $tpl    = & $this -> loadview();        //实例化视图
        $tpl -> assign("pagebar",$pbar);
        $tpl -> assign("list",$list);
        $tpl -> assign("clist",$clist);
        $tpl -> display($tplym);
    }
```

【代码解读】

该部分通过 xspic($xs)中$xs 的传入值来加载不同的显示列表页面，其中规定了不同形式下的图片显示尺寸。然后使用前文介绍过的 searchArtimage()和 getCategoryList()方法将数据输出到模板中。

2．模型

图片详情的模型只涉及了一个“业务”功能，就是根据图片 id 显示图片详情函数 getArtimageInfoByid($id)。该函数代码如下：

```
/**
* 函数名称：  获得交易图片的详细信息
* 输入参数：
    (int)$id:    图片序号
* 返回参数：
    (array):     操作成功返回记录数组，反之，返回 false
**/
function getArtimageInfoByid($id){
    $db     = & $this->db();
```

```
    $id      = intval($id);
    $query  = "SELECT * FROM tp_picinfo where picid = '$id'";
    $rt      = $db->Query($query);
    $rtarr  = $db->getrow($rt);
    if($rtarr) {
        Return $rtarr;
    }else {
        Return false;
    }
```

【代码解读】

获取图片详情功能是通过以图片 id 编号为条件，从图片详情表中获得对应行的全部内容，然后通过 getrow()函数取出结果，返回结果集。

3. 视图

图片详情页面的视图部分可以分为 3 个部分。第一个部分是显示预览大图；第二个部分是左侧的详细信息栏；第三部分是下方的“相似图片列表”。下面分别来看这 3 个部分的具体设计。

预览大图部分，核心的代码很简单，代码如下：

```
<img src="<%$image_dir%><%$info.Code%>.gif" width="290" height="396" />
```

左侧详细信息栏部分主要功能是将详细信息生成对应的模板标签，然后输出到模板中。左侧详细信息栏部分文件代码如下：

```
<table width="100%" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td class="zt top"> </td>
    <td height="30" class="zt top"><%$info.Name%></td>
  </tr>
  <tr>
    <td class="zt"> </td>
    <td height="25" class="zt">编号：<%$info.Code%> (特定使用范围版权图片)</td>
  </tr>
  <tr>
    <td class="zt"> </td>
    <td height="25" class="zt">作者：<%$info.Artist%></td>
  </tr>
  <tr>
    <td class="zt"> </td>
    <td height="25" class="zt">尺寸：<%$info.Psize%></td>
  </tr>
  <tr>
    <td class="zt"> </td>
    <td height="25" class="zt">可印刷开数：<%$info.Ysks%></td>
  </tr>
  <tr>
    <td class="zt"> </td>
    <td height="25" class="zt">版权所属：<%$info.Copy%></td>
  </tr>
  <tr>
    <td class="zt"> </td>
```

```
    <td height="25" class="zt">备注：此图只有 5 到 10M，请客户慎用 </td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
  </tr>
  <tr>
    <td class="zt"> </td>
    <td class="zt">如您对图片有更进一步的需求，<a href="#"><font color=
    "#0033FF"><b>联系我们</b></font></a></td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
  </tr>
  <tr>
    <td> </td>
    <td><table width="100%" border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td width="22%" rowspan="2"><img src="<%$image dir%>buy.gif"
        width="114" height="42" /></td>
        <td width="4%"><img src="<%$image dir%>but01.gif" width="18"
        height="20" /></td>
        <td width="74%" class="zt"><a href="?/c/artimage/act/like/cid/
        <%$info.cid%>">查看类似图片</a></td>
      </tr>
      <tr>
        <td><img src="<%$image dir%>but02.gif" width="18" height="20"/>
        </td>
        <td class="zt"><a href="?/c/add cart/picid/<%$info.Picid%>/
        name/<%$info.Name%>/artist/<%$info.Arsist%>/psize/<%$info.
        Psize%>">加入购物车</a></td>
      </tr>
    </table></td>
  </tr>
</table>
```

【代码解读】

在图片详情程序的代码中了解到，“相似图片列表”部分代码调用了另一个模板（xspiclist.html），这方便了后期的管理及样式更新。由于是在一个模板中调用另一个模板文件，所以这里使用 iframe 标签来引入新的模板文件。代码如下：

```
<iframe frameborder="0" height="110" width="100%" scrolling="no" src="?
/c/artimage/act/likeframe/cid/<%$info.cid%>"></iframe>
```

在浏览器中执行下面的链接：

```
http://localhost/pt/index.php?/c/artimage/act/show/id/4
```

执行该链接后，图片详情页面的效果如图 20.10 所示。





相似图片（以作品名称关键字列出相似图片）



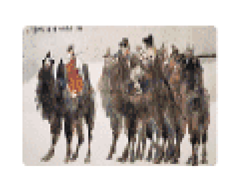



图 20.10　图片详情页面

20.5　图片收藏及订购

在前一节介绍了图片的展示和搜索流程，在这个过程中用户已经可以选到满意的图片商品，接下来就进入核心的业务流程，即图片的收藏及订购流程。本节将具体地介绍图片收藏及订购流程的相关功能的设计和代码实现。根据用户的交易流程，下面分为 3 个部分的衔接来实现用户的交易流程：

（1）图片收藏。

（2）图片购物车。

（3）订单的确认及生成。

20.5.1　图片收藏

图片收藏功能是一个非常实用的功能。用户在选择图片商品的时候可以先收藏比较满意的图片商品，挑选完毕后再到用户中心查看图片收藏夹，批量地下订单完成购买流程。创建控制器文件 c_collection.php，该文件负责处理收藏业务相关的流程。

1. 控制器

图片收藏控制的核心代码如下：

```
//是否正常读取
if(!defined('APPNAME')) {                    //权限检查
    exit("没有权限");
}
uses("error");
```

```
class c_collection extends controller {
    //控制器
    function c_collection() {
        loadModel("user_authz");                    //加载用户权限验证类
        $this->ua   = & new user_authz;
        if($this->ua->isLogin()) {                  //验证登录状态
            $this->uid  = $this->ua->getAuthzId();
            $this->uname= $this->ua->getAuthzName();
        }else{
            $tpl    = & $this -> loadview();
            $bu     = "?/c/login/";
            error::_view($tpl,'没有访问权限！',$bu);
        }
        loadModel("collection");                    //加载收藏业务模型
        $this->co = & new collection;               //实例化控制器
    }
    //默认动作
    function onMain() {
        if(GP('act')=="") {
            $this -> onlist();                      //默认加载 onlist 方法
        }
    }
    //加入收藏夹
        function onadd(){
        $this->debugon();                           //开启报错抛出
        $co     = &$this->co;
        $backurl= getBackUrl();                     //返回链接
        $picid  = GP('id');
        $rs     = $co -> addcollection($picid,$this->uid); //加入收藏夹方法
        $tpl    = & $this -> loadview();
        if($rs) {
            error:: View($tpl,"成功加入收藏夹",$backurl);
        }else {
            error::_View($tpl,"无法加入收藏夹，请与管理员联系！",$backurl);

        }
    }
```

【代码解读】

由于图片收藏功能是提供给注册用户使用的，所以在构造函数中加入用户权限验证模型（user_authz）来处理用户权限认证的问题。这里需要注意的是，图片收藏业务控制器的默认加载动作是 onlist()方法，这个方法被定义为列出全部的已收藏的图片，在后面会具体介绍。添加图片的功能由 onadd()方法来实现，实现过程是获得商品 id 编号后通过 addcollection()方法加入收藏夹。代码如下：

```
$picid  = GP('id');
$rs     = $co -> addcollection($picid,$this->uid);
```

2. 模型

图片收藏模型类（collection.php）的功能是操作图片收藏表（tp_collection）实现图片的添加（收藏）、删除等基本操作。与收藏图片相关的模型方法如下：

```
    /**
    * 函数名称：  收藏夹中图片是否已经存在
```

```
    * 输入参数：
        string  $picid      图片 id
        int     $uid        用户 id
    * 返回参数： boolen
    **/
    function is exists($picid,$uid){
        $db     = & $this->db();                    //实例化数据库连接
        $uid    = intval($uid);                     //获得用户 uid
        $query  = "select count(*) FROM tp_collection where Uid = '$uid' and
        Picid='$picid'";
        $rt     = $db->Query($query);               //执行 SQL
        $rs     = $db->getrow($rt);
        if($rs) {                                   //结果集验证
            if($rs[0]>0) {
                Return true;
            }else {
                Return false;
            }
        }else {
            Return false;
        }
    }

    /**
    * 函数说明：加入收藏夹
    * 输入参数：
        mixed   $picid  加入收藏夹的图片 id 或者图片 id 的数组
        int     $uid 用户 id
    * 返回参数： boolen
    **/
    function addcollection($picid,$uid){            //加入收藏夹
        $db     = & $this->db();                    //实例化数据库连接
        $uid    = intval($uid);
        $picid  = intval($picid);                   //获得图片编号
        if(!$this->is_exists($picid,$uid)) {        //判断图片是否已经存在
            $query  = "insert into tp collection(id,Uid,Picid,adddate)
            values('','".$uid."','".$picid."',NOW())";
        }
        $rt = $db->Query($query);                   //执行 SQL
        Return $rt;                                 //返回操作结果
    }
}
    * 函数名称： 删除收藏夹的图片记录
    * 输入参数：
        mixed   $id     收藏夹 id 或者收藏夹 id 的数组
        int     $uid    用户 id
    * 返回参数： boolen
    **/
    function delcollection($id,$uid){               //删除收藏夹的图片记录
        $db     = & $this->db();
        $uid    = intval($uid);                     //获得 uid
        if(is array($id)) {
            $val    = implode(',',$id);             //截取 id 串
            $query  = "delete FROM tp collection where id in ($val) and Uid
            = '$uid'";1
        }else {
```

```
            $id     = intval($id);  //获得收藏夹 id 或者收藏夹 id 的数组
            $query  = "delete FROM tp collection where id='$id' and Uid =
            '$uid'";
        }
        $rt     = $db->Query($query);//执行 SQL
        Return $rt;
    }
```

【代码解读】

以上代码涉及了在图片收藏业务模型的 3 个功能（收藏夹中图片是否已经存在、加入收藏夹的图片 id 或者图片 id 的数组、删除收藏夹的图片记录）。在添加和删除两个操作时，都需要先检查图片收藏表中是否存在被选中的 id，这样就可以避免因逻辑冲突造成的程序异常。

以上介绍了图片收藏的代码实现，由于图片收藏动作中不涉及视图的展现，所以在控制器中没有加载视图。

在浏览器中执行下面的链接：

```
http://localhost/pt/?/c/collection/act/add/id/5
```

执行该链接后提示图片收藏成功的页面效果，如图 20.11 所示。



图 20.11　图片收藏

20.5.2　图片购物车

购物车的实现过程是本节的重点内容，购物车的作用是用来临时存储用户的购物的商品信息，用户可以修改购物车中的商品数量、移除购物车中的指定的商品、清空购物车。本节通过 PHP 中的 Session 函数，来实现对用户购物过程中的商品信息的存储。Session 是存储在服务器端的会话，适合存储本次会话中的临时数据，当本次访问结束，Session 会自动被注销，这点希望读者注意。

下面介绍在 PHP 中通过 MVC 模式实现图片购物车流程。创建控制器文件 add_cart.php，该文件负责处理与图片购物车相关的业务流程。

1．控制器

图片购物车控制的核心代码如下：

```
//加载控制器基类
if(!class_exists('controller')) {
```

```
    uses('controller');
}
class add_cart extends controller {
    function __construct() {
        //加载核心购物类
        loadModel('class_Cart');
        //初始化 session
        @session_start();
        //实例化
        $this->carts= new x_cart();
    }

    function onMain() {
        if(empty($_REQUEST['act'])) {
            $this -> ondisplay();       //默认 action
        }
    }

    function ondisplay(){
        $id     =GP('picid');
        $name   =GP('name');
        $artist =GP('artist');
        $psize  =GP('psize');
        //判断是否传递值过来，有值则添加到购物车
        if(is_array($id)==True)
        {
            //获得 POST 过来的商品信息数组
            $arr=GP('arr');
            for($i=0;$i<count($id);$i++){
                //加入购物车
                $this->carts->add_item(array($id[$i],$name[$i],$artist
                [$i],$psize[$i]),1);
            }
            }elseif(isset($id) and is_array($id)==False){
                //加入购物车
                    $this->carts->add_item(array("{$id}","{$name}",
                    "{$artist}","{$psize}"),1);
                }else{
                    exit("出错，没有添加商品<a href=index.php>返回</a>");
                }
                    header("location: http://localhost/pt/index.php?/
                    c/cart/");
            }
        }
```

【代码解读】

图片购物车控制器的默认动作，该动作是执行将商品添加到购物车中的方法 ondisplay()，这里提供了批量添加到购物车中的功能。因此，在执行添加方法业务类 add_item()之前，需要判断本次添加的商品数量，即商品的 id 编号数量，添加成功后跳转到购物车的查看页面。跳转地址代码如下：

```
header("location: http://localhost/pt/index.php?/c/cart/");
```

cart.php 控制器文件实现的功能是通过读取购物车中的商品 Session 会话变量，然后按照规则显示出来。该控制器与 add_cart.php 控制器功能不同的是，add_cart.php 控制器中的方法是处理添加写入 Session 会话变量这个过程的，通俗的说，就是往购物车中放商品的

过程；而 cart.php 是来处理购物车中信息展示的。

cart.php 控制器文件的核心代码如下：

```
if(!class exists('controller')) {
    uses('controller');                         //控制器基类
}
class cart extends controller {
    function __construct() {
        loadModel('class Cart');                //加载核心购物类
        @session_start();                       //初始化 session
        $this->carts= new x cart();             //实例化
    }
//默认动作
    function onMain() {
        //$id   = GP('id');
        if(empty($_REQUEST['act'])) {
            $this -> ondisplay();               //默认 action
        }
    }
//模型层
    function ondisplay(){
        $smarty = & $this -> loadview();
        $arr=$this->carts->get_all();           //判断购物车的状态是否有物品
        if(empty($arr)){
            $input array=array();               //数组方式输出购物车中物品
            }else{
            $input array=$this->carts->get all();
        }
    /****************分页处理***********************/
    /*
    * 输入参数：
        (int) $tr   总记录数
        (int) $pz   页面大小
        (int) $cp   当前页数
        (str) $lp   链接参数
        (int) $st   分页跨度
    * 返回参数：
        array(  'totalrows',                    //总记录数
                        'pagesize',             //页面大小
                        'stride',               //跨度大小
                        'currentpage',          //当前页数
                        'linkparam',            //链接参数
                        'totalpages',           //总页数
                        'startrow',             //开始行数
                        'endrow',               //结束行数
                        'prevpage',             //上一页页数
                        'nextpage',             //下一页页数
                        'stride start',         //跨度开始页数
                        'stride_end',           //跨度结束页数
                        'stride_array'          //跨度内所有页数的数组
                        );
    **/
        uses("pagebar");                        //基础分页类
        $pb     = new pagebar;
        $pz     = 10;                           //页面大小，单页显示条数
```

```
        $lp     = url('cart','display',array());
        $tr     = $total=count($input_array);              //获得总记录数
        $pbar   = $pb-> execute($tr,$pz,$lp);
        $i = (empty($_GET['page'])) ? '0' : $_GET['page']-1;
        $page=array_chunk($input_array, $pz, true);        //翻页数组
        $smarty -> assign("records",$page[$i]);            //分页数组
        $smarty -> assign("amount",count($input_array));   //总条数
        $smarty -> assign("pagebar",$pbar);
        $smarty -> assign("state",$arr);
        $smarty -> display('shopping.html');
    }
}
```

【代码解读】

cart.php 控制器通过 carts->get_all();判断购物车的状态是否有物品，当购物车中的商品过多时要做分页处理，然后加载购物车显示视图。

2. 模型

购物车业务核心类 class_Cart.php 文件，负责处理操作购物车的具体动作，业务流程就是这些动作的组合。购物车业务核心类代码如下：

```
<?php
class carts {
    var $items;
    var $total;

    /**
     *构造函数
     */
    function carts($varname = 'cart') {     //默认购物车 session 名字为 cart
        $this->items = &$_SESSION [$varname];  //引用，绑定变量变化
    //$this->items = &$varname;                //引用，绑定变量变化
    }
    /**
     *添加一个商品
     *$item 数字或数组(依次是 id、商品名、零售价、销售价<可选>)
     *$num 数量
     */
    function add_item($item, $num) {
        $id = $this->add_merchandise ( $item );
        $this->items [$id] ['quantity'] += $num;
        $this->items [$id] ['discount'] = ($this->items [$id] ['retail
        price'] - $this->items [$id] ['price']) * $this->items [$id] ['
        quantity'];
        //更新折扣
        $this->items [$id] ['subtotal'] = $this->items [$id] ['quantity']
        * $this->items [$id] ['price'];          //更新小计
        return true;
    }
    /**
     *减商品
     */
    function remove_item($id, $num) {
        if ($this->is_item ( $id ) == false)
            return false;                        //检测是否在购物车中的物品
        if ($this->items [$id] ['quantity'] > $num) {
```

```
                                    //如果购物车中的物品大于要减的数量
            $this->items [$id] ['quantity'] -= $num; //从购物车中减出
            $this->items [$id] ['discount'] = ($this->items [$id] ['retail_
            price'] - $this->items [$id] ['price']) * $this->items [$id]
            ['quantity'];                   //更新折扣
            $this->items [$id] ['subtotal'] = $this->items [$id] ['quantity']
            * $this->items [$id] ['price'];//更新小计
            return true;
        } elseif ($this->items [$id] ['quantity'] == $num) {
                                    //如果购物车中的物品等于要减的数量
            unset ( $this->items [$id] );  //删除购物车中该物品的所有资料
            return true;
        } else {
            return false;
        }
    }
    /**
     *编辑商品数量
     */
    function edit item($id, $num) {
        if ($this->is item ( $id ) == false)
            return false;               //检测是否在购物车中的物品
        if ($num > 0) {                 //保存 0 件以上的商品放在购物车里
            $this->items [$id] ['quantity'] = $num;
            $this->items [$id] ['discount'] = ($this->items [$id] ['retail
            price'] - $this->items [$id] ['price']) * $this->items [$id]
            ['quantity'];               //更新折扣
            $this->items [$id] ['subtotal'] = $this->items [$id] ['quantity']
            * $this->items [$id] ['price']; //更新小计
            return true;
        } elseif ($num == 0) { //如果是 0 件
            unset ( $this->items [$id] ); //删除购物车中该物品的所有资料
            return true;
        } else {
            return false;
        }
    }

    /**
     *删除购物车中该物品的所有资料
     */
    function dele_item($id) {
        if ($this->is item ( $id ) == false)
            return false;
        unset ( $this->items [$id] );
        return true;
    }

    /**
     *检测购物车中该物品是否存在
     */
    function is_item($id) {
        return is array ( $this->items [$id] );
    }
    /**
     *清空购物车
     */
    function empty_cart() {
```

```
        $this->items = NULL;        //清空购物车
    }
    /**
     *预留方法接口，根据 ID 取得商品信息源
     */
    function get merchandise($id) {
        return false;
    }
    /**
     *统计购物车
     */
    function sum item() {
        if (! is array ( $this->items ))
            return false;
        foreach ( $this->items as $value ) {
            $total ['quantity'] += $value ['quantity'];    //总数量
            $total ['discount'] += $value ['discount'];    //总折扣
            $total ['monny'] += $value ['subtotal'];       //总金额
        }
        $this->total = $total;
        return $total;
    }
}
//获取购物车中的内容
function get all($value = '') {
    if (isset ( $this->items )) {
        $value = $this->items;                  //获得购物车中的内容
        return $value;
    } else {
        return false;                           //获取失败
    }
}
//获得购物车中物品数组 id
function all_id($arr = '') {
    if (isset ( $this->items )) {
        foreach ( $this->items as &$value ) {
            $arr [] = $value ['id'];
        }
        return $arr;
    } else {
        return false; //获取失败
    }
}

?>
```

【代码解读】

上面的代码中包括控制购物车的常见动作，这些动作是具体完成某一项功能的方法（添加一个商品、减去商品、编辑商品数量、删除购物车中该物品的所有资料、检测购物车中该物品是否存在、清空购物车、统计购物车、获取购物车中的内容，获得购物车中物品数组 id 等），这些方法的组合完成用户使用购物车的完整流程。

3．视图

由于将图片添加进购物车的过程实际上是对 Session 会话变量的操作，不涉及用户视图的部分，这里着重介绍购物车中内容的展示视图。显示购物车中内容的核心视图代码

如下：

```
<table width="90%" border="0" align="center" cellpadding="1"
    cellspacing="1" bgcolor="#999999">
    <tr>
        <td width="6%" height="30" align="center" bgcolor="#CCCCCC"
            class="zt_2">序号</td>
        <td width="10%" align="center" bgcolor="#CCCCCC" class="zt_2">图
        片编号</td>
        <td width="37%" bgcolor="#CCCCCC" class="zt_2">作品名称</td>
        <td width="10%" align="center" bgcolor="#CCCCCC" class="zt_2">作
        者</td>
        <td width="11%" align="center" bgcolor="#CCCCCC" class="zt_2">尺
        寸（CM）</td>
        <td width="8%" align="center" bgcolor="#CCCCCC" class="zt_2">操作
        </td>
    </tr>
    <%counter start=0 skip=1 print=false%> <%foreach key=okey item=oitem
    from=$records%>
    <tr>
        <td height="30" align="center" bgcolor="#FFFFFF" class="zt">
        <%counter%></td>
        <td align="center" bgcolor="#FFFFFF" class="zt"><%$oitem.id%></td>
        <td bgcolor="#FFFFFF" class="zt"><%$oitem.name%></td>
        <td align="center" bgcolor="#FFFFFF" class="zt"><%$oitem.retail
        price%></td>
        <td align="center" bgcolor="#FFFFFF" class="zt"><%$oitem.price%>
        </td>
        <td align="center" bgcolor="#FFFFFF" class="zt"><a href=#
            onclick=del('<%$oitem.id%>',"?/c/Del/id/","确认删除!")>删除
            </a></td>
    </tr>
    <%/foreach%>
</table>
```

当用户将选中的商品添加进购物车后，浏览器将跳转到下面的链接：

```
http://localhost/pt/index.php?/c/cart/
```

执行该链接后将显示购物车中的商品信息，效果如图 20.12 所示。



图 20.12　购物车

20.5.3　订单处理流程

用户已经将需要购买的物品放入了购物车中，下面介绍用户如何将购车中的物品保存

为一个订单，这个过程就是一个图片商品的购买过程。在生成有效订单后，系统就会根据订单的信息执行线下的“发货”流程了。在技术层面，订单的处理过程实际上是一个标准的事务处理，读者在本节中需要处理 PHP 实现事务处理的过程。

下面介绍在 PHP 中 MVC 模式下的订单处理。首先创建控制器文件 process.php，该文件负责订单流程的处理。

1. 控制器

process.php 控制器文件的核心代码如下：

```
if(!class exists('controller')) {
    uses('controller');
}
class process extends controller {
    function __construct() {
        loadModel('class Cart');                //加载核心购物类
        @session_start();                       //初始化 session
        $this->carts= new x cart();             //实例化
        //判断用户是否登录
        loadModel("user_authz");                //判断用户是否登录类
        $this->ua   = & new user authz;
        if($this->ua->isLogin()) {
            $this->uid  = $this->ua->getAuthzId();     //用户 id
            $this->uname= $this->ua->getAuthzName();
                            //当前用户名（获取当前用户信息是根据用户名来获得的）
        }else{//报错抛出
            $tpl    = & $this -> loadview();
            $bu     = "?/c/login/";
            error::_view($tpl,'没有访问权限！',$bu);
        }
    }
    function onMain() {
        if(empty($ REQUEST['act'])) {
            $this -> onstep1();                 //默认 action
        }
    }
    //订单流程第一步(输入送货信息)
    function onstep1(){
        $uinfo  = $this->ua->getuserByName($this->uname);//获得当前用户信息
        $smarty = & $this -> loadview();        //加载模板配置
        $smarty ->  assign ('userinfo',$uinfo);//当前用户信息
        var dump ($uinfo);
        $smarty ->  display('shopping_step2.html');
    }
    //订单流程第二步（订单确认）
    function onstep2(){
        $arr['name']    =GP('Name');            //用户名
        $arr['address']=GP('Address');          //发货地址
        $arr['zip'] =GP('Zip');                 //邮编
        $arr['tel'] =GP('Tel');                 //电话
        $arr['fax'] =GP('Fax');                 //传真
        $smarty = & $this -> loadview();        //加载 smarty 配置
```

```
        $smarty -> assign('arr',$arr);              //订单数组
        //获得全部商品
        $input array=$this->carts->get all();
        $smarty->assign("records",$input_array);    //分页数组
        $smarty ->display('shopping step3.html');
    }
}
```

【代码解读】

订单处理流程是一个用户私有的流程，因此首先通过 user_authz 类判断用户的登录状态，然后用 getAuthzId()方法和 getAuthzName()方法获得用户的 id 和用户名信息。代码如下：

```
loadModel("user authz");//判断用户是否登录类
$this->ua   = & new user authz;
if($this->ua->isLogin()) {
//用户 id
$this->uid  = $this->ua->getAuthzId();
//当前用户名（获取当前用户信息是根据用户名来获得的）
$this->uname= $this->ua->getAuthzName();
}
```

获得用户信息后生成订单的过程分为两个步骤，即 onstep1()方法和 onstep2()方法。作用分别是输入送货信息和确认订单。这个过程中用到获得当前用户信息 getuserByName()方法和获得购物车中全部商品的 get_all()方法。

2．模型

订单处理流程中用的业务模型最主要的就是 get_all()方法，它属于购物车业务模型类中的扩展方法，用于获得购物车中全部的内容。该方法的内容如下：

```
//获取购物车内容类
function get all($value=''){
    if(isset($this->items)){
        $value=$this->items;        //获得购物车中的内容
        return $value;              //返回购物车内容数据
        }else{
        return false;               //获取失败
    }
}
```

【代码解读】

该方法中的$this->items 是类中定义的购物车中内容的数组变量。该变量由 Session 数组构成，在商品添加进购物车时，将物品信息“写入”Session 数组中。

3．视图

订单处理流程中页面展示分为 3 个步骤完成：

（1）核对“送货信息”（shopping_step2.html）。

（2）确认“订单信息”（shopping_step3.html）。

（3）“订单提交成功”（shopping_step4.html）。

核对送货信息（shopping_step2.html）的核心是验证用户注册信息表单。该文件表单的核心代码如下：

```
<form name="Default" method="post" action="?/c/process/act/step2">
<table width='100%' height='0' border='0' cellpadding='0'
    cellspacing='10'>
    <tr>
        <td align='left'><input type='submit' value='配送到这个地址'><br />
        <br />
        <span class="zt"> 姓名：<%$userinfo.U_truename%><br />
        地址：<%$userinfo.U_address%><br />
        邮编：<%$userinfo.U_postcode%><br />
        电话：<%$userinfo.U_phone%><br />
        传真：<%$userinfo.U_fax%><br />
        </span><br />
        <input type="hidden" name="Name" value="<%$userinfo.U_truename%>"/>
        <input type="hidden" name="Address" value="<%$userinfo.U_address%>"/>
        <input type="hidden" name="Zip" value="<%$userinfo.U_postcode%>"/>
        <input type="hidden" name="Tel" value="<%$userinfo.U_phone%>"/>
        <input type="hidden" name="Fax" value="<%$userinfo.U_fax%>"/><br />
        <input type='button' value='修改' onClick="show()">   </td>
    </tr>
</table>
</form>
```

执行下面的链接，送货信息核对的页面如图 20.13 所示。

```
http://localhost/pt/index.php?/c/process/
```

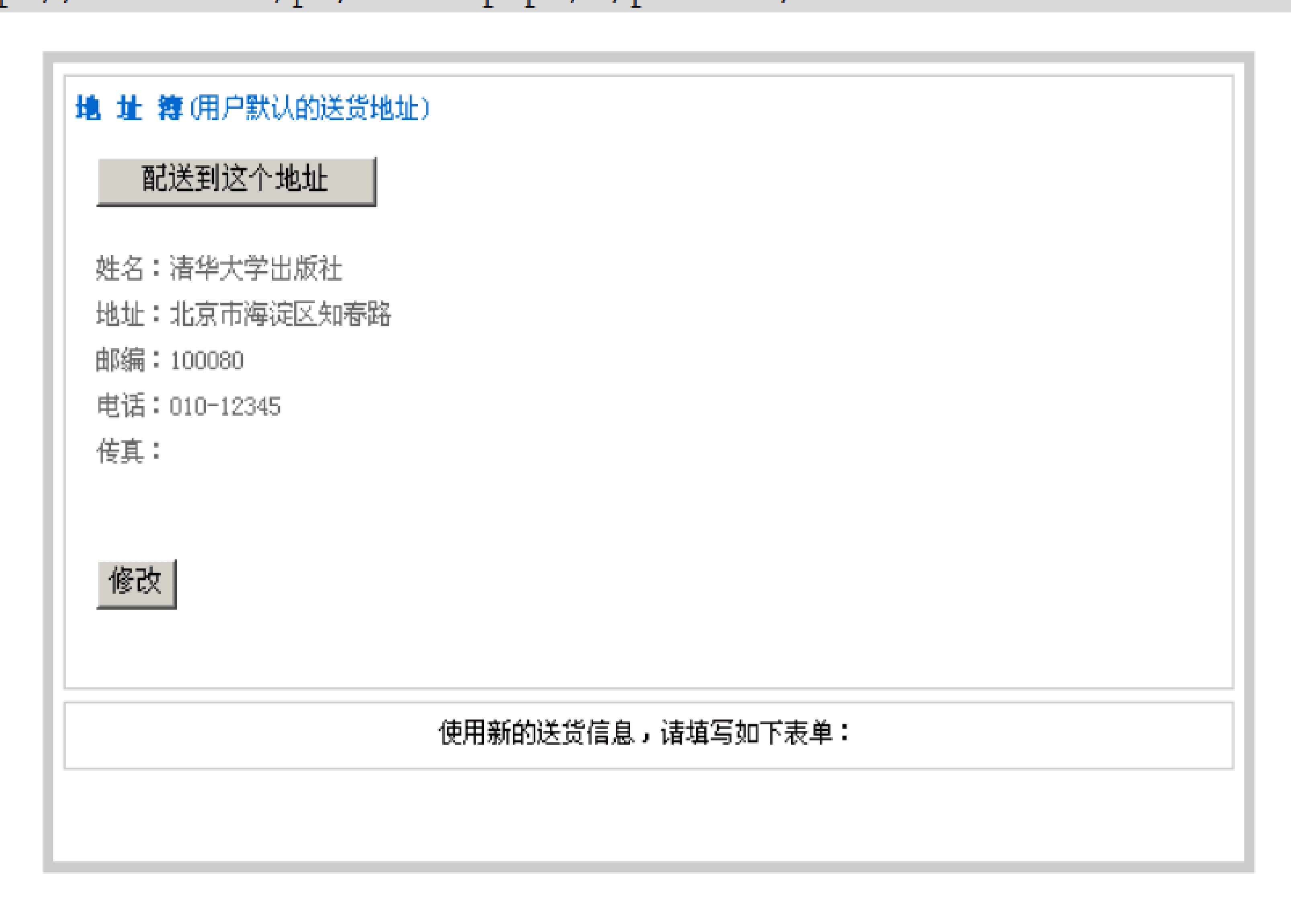


图 20.13　核对送货信息

当用户注册信息出错的时候，可以通过单击界面上的“修改”按钮，修改订货的用户信息。该部分的表单代码如下：

```
<form name="info" method="post" action="?/c/process/act/step2">
<table width="100%" border="0" cellpadding="0" cellspacing="1"
```

```
    bgcolor="#e3e3e3" align="center">
    <tr align="left" bgcolor="#f6f6f6">
        <td height="25" colspan="3"><font color="#ff0000">*</font><span
            class="zt">为必填选项 (输入完毕后,请点击按钮"配送至这个地址")</span>
            </td>
    </tr>
    <tr bgcolor="#ffffff">
        <td height="35" colspan="2" align="right"><span class="zt">收货人
        姓名：</span></td>
        <td width="427" align="left"><input name="Name" type="text"
            size="10" maxlength="50"> <font color="#ff0000">*</font></td>
    </tr>
    <tr bgcolor="#f6f6f6">
        <td height="35" colspan="2" align="right"><span class="zt">地址：
        </span></td>
        <td height="25" align="left"><font color="#ff0000"> <input
            name="Address" type="text" size="50" maxlength="500" /> <font
            color="#ff0000">*</font></span></font></td>
    </tr>
    <tr bgcolor="#ffffff">
        <td height="35" colspan="2" align="right"><span class="zt">邮政编
        码：</span></td>
        <td height="25" align="left"><input name="Zip" type="text"
            size="6" maxlength="50" /> <font color="#ff0000">*<font
            color="#999999" class="zt"> 请正确填写邮政编码，以免延误您的订单送
            达时间。</font></font></td>
    </tr>
    <tr bgcolor="#f6f6f6">
        <td height="35" colspan="2" align="right"><span class="zt">电话：
        </span></td>
        <td height="25" align="left"><input name="Tel" type="text"
            size="12" maxlength="50" /> <font color="#ff0000">*</font>
            <font
            color="#999999" class="zt">（请填写区号）</font></td>
    </tr>
    <tr bgcolor="#ffffff">
        <td height="35" colspan="2" align="right"><span class="zt">传真：
        </span></td>
        <td height="25" align="left"><input name="Fax" type="text"
            size="10" maxlength="50" value="" /> <span class="zt">（传真和
            电话至少有一项必填）</span></td>
    </tr>
    <tr>
        <td width="146" height="75"></td>
        <td width="13" height="75"></td>
        <td height="75" align="left"><input type="submit" value="配送到这
        个地址"></td>
    </tr>
</table>
</form>
```

单击“修改”按钮修改用户注册信息的效果，如图 20.14 所示。



图 20.14　修改用户注册信息

订单确认部分的核心代码如下：

```
<table width="60%" height="111" border="5" cellpadding="5"
    cellspacing="5" bordercolor="#CCCCCC" align="center">
    <tr>
        <td height="25" bgcolor="#DBE3F7" class="ti"><strong>收货人信息
        </strong>- <span class="zt">修改</span></td>
    </tr>
    <tr>
        <td>
        <table width="90%" border="0" align="center" cellpadding="0"
            cellspacing="0">
            <tr>
                <td width="10%" height="26" class="zt">姓名：</td>
                <td width="90%" class="zt"><%$arr.name%></td>
            </tr>
            <tr>
                <td height="26" class="zt">地址：</td>
                <td class="zt"><%$arr.address%></td>
            </tr>
            <tr>
                <td height="26" class="zt">邮编：</td>
                <td class="zt"><%$arr.zip%></td>
            </tr>
            <tr>
                <td height="26" class="zt">电话：</td>
                <td class="zt"><%$arr.tel%></td>
            </tr>
            <tr>
```

```
                <td height="26" class="zt">传真：</td>
                <td class="zt"><%$arr.fax%></td>
            </tr>
        </table>
        </td>
    </tr>
    <tr>
        <td height="25" bgcolor="#DBE3F7"><span class="ti"><strong>作品信
        息</strong></span>
        <span class="ti">-</span> <span class="zt">修改</span></td>
    </tr>
    <tr>
        <td>
        <table width="90%" border="0" align="center" cellpadding="1"
            cellspacing="1" bgcolor="#999999">
            <tr>
                <td width="6%" height="30" align="center" bgcolor=
                "#CCCCCC"
                    class="zt_2">序号</td>
                <td width="10%" align="center" bgcolor="#CCCCCC" class=
                "zt_2">图片编号</td>
                <td width="57%" bgcolor="#CCCCCC" class="zt_2">作品名称</td>
                <td width="13%" align="center" bgcolor="#CCCCCC" class=
                "zt_2">作者</td>
                <td width="14%" align="center" bgcolor="#CCCCCC" class=
                "zt_2">尺寸（CM）</td>
            </tr>
            <%counter start=0 skip=1 print=false%> <%foreach key=okey
            item=oitem
            from=$records%>
            <tr>
                <td height="30" align="center" bgcolor="#FFFFFF" class=
                "zt"><%counter%></td>
                <td align="center" bgcolor="#FFFFFF" class="zt"><%$oitem.
                id%></td>
                <td bgcolor="#FFFFFF" class="zt"><%$oitem.name%></td>
                <td align="center" bgcolor="#FFFFFF" class="zt"><%$oitem.
                retail_price%></td>
                <td align="center" bgcolor="#FFFFFF" class="zt"><%$oitem.
                price%></td>
            </tr>
            <%/foreach%>
        </table>
        </td>
    </tr>
    <tr>
        <td height="25" bgcolor="#DBE3F7"><STRONG class="ti">备注</STRONG>
        <span class="ti">-</span> <span class="ti">您有什么需要说明的请在备注
        里填写</span></td>
    </tr>
    <tr>
        <td>
        <table width="90%" border="0" align="center" cellpadding="0"
            cellspacing="0">
            <tr>
                <td height="30"><textarea name="remark" cols="50" rows="5"
                    id="remark"></textarea></td>
            </tr>
        </table>
```

```
        </td>
    </tr>
</table>
```

单击“配送到这个地址”按钮后，页面跳转到订单确认的页面，这步已经生成了完整的用户订单。效果如图 20.15 所示。



图 20.15　订单确认

当用户确认订单信息无误后，单击“提交订单”，弹出订单提交成功页面。效果如图 20.16 所示。



图 20.16　订单提交成功

20.6　用户管理中心

在电子交易平台一类的系统中，通常会设置用户管理中心这一功能，该功能是用来提供给用户自主管理用户基本信息。图片交易平台模块中，用户管理中心主要负责管理用户

身份验证、订单管理、收藏夹管理和出版社信息管理。用户可以通过 Web 界面直观地修改用户的基本信息，以及处理购买流程中的相关信息。

下面开始分别介绍用户管理中心的 4 项基本管理功能。本节中的用户管理中心是对前面介绍的业务功能的用户自主管理，读者注意和前文介绍的具体的业务功能的对应。

20.6.1　用户登录验证

用户登录验证功能是对用户的身份进行验证，验证通过后，系统会记录该用户的登录信息和状态，并在用户整个浏览过程中保持身份信息和相应的权限。用户登录验证功能是所有电子交易平台中都不可或缺的一项基本功能，实现的原理在前面的模块中曾经介绍过。在学习本节时可以复习相关的实现原理，重点内容是了解在 MVC 框架中的组织模式。

创建控制器文件 c_user.php，下面介绍控制器部分的设计。

1．控制器

用户认证控制器（c_user.php）的代码如下：

```
<?
//是否正常读取
if(!defined('APPNAME')) {                                //权限验证
    exit("没有权限");
}
uses("error");
class c_user extends controller {
    //构造函数
    function c_user() {
        loadModel("user_authz");                         //加载用户认证模型
        $this->ua    = & new user_authz;                 //实例化用户验证类
        if($this->ua->isLogin()) {
            $this->uid  = $this->ua->getAuthzId();
            $this->uname= $this->ua->getAuthzName();
        }else{
            $tpl     = & $this -> loadview();            //实例化视图类
            $bu      = "?/c/login/";
            error::_view($tpl,'没有访问权限！',$bu);     //异常报错抛出
        }
        loadModel("user");                               //加载用户基类
        $this->u = & new user;                           //实例化用户基类
    }
    //默认动作
    function onMain() {
        if(GP('act')=="") {
            //加载默认动作
        }
    }
    //密码合法性验证
    function wrongpwd($pwd) {
        if(ereg("[0-9a-zA-Z]{6,12}",$pwd)) {             //正则匹配密码规则
            Return false;
        }else {
            Return true;
        }
```

```
    }
}
```

【代码解读】

用户认证控制的实现流程如下：

（1）通过构造函数加载用户认证权限类 user_authz。

（2）判断用户当前的状态，如果已经登录，调用 getAuthzId()和 getAuthzName()方法，获得用户 id 和用户名信息；如果没有登录，加载用户登录界面。

2. 模型

用户业务核心类（user_authz.php）文件的代码如下：

```
<?php
//是否正常读取
if(!defined('APPNAME')) {                          //权限验证
    exit("没有权限");
}
uses("authz","error");
class user_authz extends model{
    var $authz_id   ='webuserid';                  //授权编号名称
    var $authz_name ='webusername';                //授权用户名称
    /**
    * 函数说明： 根据 name 获取用户信息
    * 输入参数： string $name 用户名
    * 返回参数： void
    **/
    function getUserByName($name){                 //列出所有
        $db     = $this->db();
        $query  = "SELECT * FROM tp_users where u_name = '$name'";
        $rs     = $db->Query($query,$startrow,$endrow);
        $rsarr  = $db->getRow($rs);                //获得用户信息结果集
        if($rsarr) {
            Return $rsarr;                         //返回用户信息结果集
        }else {
            Return false;
        }
    }
    /**
    * 函数说明： 用户登录
    * 输入参数：
        string $u 登录用户
        string $p 登录密码
    * 返回参数： boolen
    **/
    function Login($u,$p) {
        $uarr   = $this->getUserByName($u);
        $p      = md5($p);                         //MD5 加密
        if(empty($uarr['U_password'])) {           //密码设置验证
            error:: set("提交用户名不存在！");
            Return false;
        }elseif($uarr['U_password'] == $p) {
            authz::_register($this->authz_id,$uarr['U_id']);
            authz::_register($this->authz_name,$uarr['U_name']);
            Return true;
```

```
        }else {
            error::_set("提交密码错误！");        //报错抛出
            Return false;
        }
    }
    /**
    * 函数说明： 用户退出
    * 输入参数： void
    * 返回参数： void
    **/
    function Logout() {
        authz::_destroy($this->authz_id);       //销毁用户 id
        authz::_destroy($this->authz_name);     //销毁用户注册名
    }
    /**
    * 函数说明： 判断用户是否登录
    * 输入参数： void
    * 返回参数： boolen
    **/
    function isLogin() {
        $rt     = authz::_is($this->authz_id); //用户 id 注册
        if($rt) {
            $rt = authz::_is($this->authz_id); //用户 id 注册
        }
        if($rt) {
            Return true;
        }else {
            error::_view("尚未登录，请先登录！"); //报错抛出
            Return false;
        }
    }
    /**
    * 函数说明： 获得授权用户值
    * 输入参数： void
    * 返回参数： void
    **/
    function getAuthzName() {
        Return authz::_get($this->authz_name);
    }
    /**
    * 函数说明： 获得授权编号值
    * 输入参数： void
    * 返回参数： void
    **/
    function getAuthzId() {
        Return authz::_get($this->authz_id);    //返回认证 id
    }
}
?>
```

【代码解读】

用户登录验证的核心是通过验证会员资料表（tp_users）中的指定记录（如用户名和密码）然后注册登录信息。代码如下：

```
authz::_register($this->authz_id,$uarr['U_id']);
authz::_register($this->authz_name,$uarr['U_name']);
```

3．视图

由于用户登录验证的过程都是在“后台”完成的，因此用户看到的界面部分只涉及登录框视图。其中验证表单代码如下：

```
<form action="?/c/login/act/login" method="post">
<table width="44%" border="0" align="center" cellpadding="0"
    cellspacing="0">
    <tr>
        <td width="47%" height="30" align="right">
        <p class="news2">用户名：</p>
        </td>
        <td width="2%"> </td>
        <td width="51%"><input name="user" type="text" size="30" /></td>
    </tr>
    <tr>
        <td height="30" align="right" class="news2">密码：</td>
        <td> </td>
        <td><input name="pwd" type="password" size="30" /></td>
    </tr>
    <tr>
        <td align="right"> </td>
        <td> </td>
        <td valign="middle">
        <table width="100%" border="0" cellpadding="0" cellspacing="0">
            <tr>
                <td width="64%"><input type="image" src="<%$image_
                dir%>login.gif"
                width="118" height="44" /></td>
                <td width="36%" class="zt 2"> </td>
            </tr>
        </table>
        </td>
    </tr>
    <tr>
        <td align="right"> </td>
        <td> </td>
        <td> </td>
    </tr>
    <tr>
        <td align="right"> </td>
        <td> </td>
        <td class="zt">如有任何疑问请与我们联系 010-1234567</td>
    </tr>
    <tr>
        <td align="right"> </td>
        <td> </td>
        <td><input name="backurl" type="hidden" size="30" /></td>
    </tr>
</table>
</form>
```

当用户点击用户登录链接 http://localhost/pt/?/c/login/，效果如图 20.17 所示。

20.6.2　订单管理

用户通过订单管理模块管理已下单的数据，系统管理员会处理用户确认的订单（下线

发货等）。该模块完成的主要功能是用户自主核对订单信息，并在发货后跟踪订单的处理状态，同时可以打印指定的订单和合同副本等凭证。



图 20.17　用户登录验证

创建控制器文件 c_order.php，该控制器负责控制订单管理部分的流程。下面介绍控制器部分的设计。

1．控制器

订单控制器代码如下：

```
<?
//是否正常读取
if(!defined('APPNAME')) {                                      //权限验证
    exit("没有权限");
}
uses("error");
class c_order extends controller {
    //订单控制器
    function c_order() {
        loadModel("user_authz");                               //加载用户验证类
        $this->ua    = & new user_authz;
        if($this->ua->isLogin()) {
            $this->uid  = $this->ua->getAuthzId();
            $this->uname= $this->ua->getAuthzName();           //获得用户名
        }else{
            $tpl     = & $this -> loadview();                  //实例化视图
            $bu      = "?/c/login/";
            error::_view($tpl,'没有访问权限！',$bu);            //异常报错抛出
        }
        loadModel("order");                                    //加载订单模型类
        $this->o = & new order;                                //实例化订单类
    }
//默认动作
    function onMain() {
        if(GP('act')=="") {
            $this -> onlist();                             //默认加载 onlist()方法
        }
    }
//订单列表
    function onlist($message='') {
        uses("pagebar");                                   //实例化数据库连接
        $o       = &$this->o;
        $ord     = GP("ord")?GP("ord"):1;                  //获得订单号
```

```
        $uinfo  = $this->ua->getuserByName($this->uname);  //获得用户名
        $pb     = new pagebar;
        $pz     = 10;
        $lp     = url('order','list',array("ord"=>$ord));
        $tr     = $o->getUserOrderCount($this->uid);
        $pbar   = $pb->_execute($tr,$pz,$lp);
        $pbar['startrow']   = 0;                        //起始记录
        $pbar['endrow']=10;                             //跨度
        //获得用户全部订单的列表
        $list   =$o->getUserOrderList($this->uid,$ord,$pbar['startrow'],
        $pbar['endrow']);
        $tj = $o->tjUserOrderInfo($this->uid);          //统计完成的订单情况
        $tpl    = & $this -> loadview();
        $tpl -> assign("pagebar",$pbar);
        $tpl -> assign("orderurl",url('order','list',array("page"=>
        GP('page'))));                                  //订单返回链接
        $tpl -> assign("tname",$uinfo['U_truename']);
        $tpl -> assign("currentdate",date('Y-m-d'));//格式化当前时间标签
        $tpl -> assign("completeorder",$tj);
        $tpl -> assign("message",$message);
        $tpl -> assign("list",$list);
        $tpl -> display('infor_order.html');
    }
//订单生成
    function onsubmit(){
        $this->debugon();
        $tpl    = & $this -> loadview();
        $backurl    = getBackUrl();                     //提交后返回链接地址
        //获得该订单的用户资料信息
        $name   = GP('name');                           //用户名
        $phone  = GP('phone');                          //电话号码
        $fax        = GP('fax');                        //传真号码
        $postcode   = GP('postcode');                   //邮编
        $address    = GP('address');                    //通讯地址
        $remark     = GP('remark');
        if(empty($name)||empty($phone)||empty($postcode)||empty($address)) {
            error::_View($tpl,"提交生成订单时，联系人、电话、邮编、地址不能为空！
            ",$backurl);
        }
        loadModel('class_Cart');                        //加载购物车类
        session_start();
        $cart   = new x_cart();                         //实例化购物车
        $goods  = $cart->all_id();                      //获得购物中商品
        $o      = &$this->o;
        $code   = $o -> addOrder($name,$phone,$fax,$postcode,$address,
        $remark,$this->uid,$goods);
        if($code) {
            $cart->empty_cart();                        //清空购物车
            $tpl -> assign("ordercode",$code);
            $tpl -> display('shopping_step4.html');
        }else {
            error::_View($tpl,"无法生成订单，请与管理员联系！",$backurl);
        }
    }

    function oncancel() {
        $o      = &$this->o;
        $backurl= getBackUrl();                         //获得返回链接地址
```

```
        $tpl    = & $this -> loadview();
        if(!GP('id')) {
            error:: View($tpl,"参数提交错误，请与管理员联系！",$backurl);
        }
        $rs     = $o -> resetOrder(GP('id'));      //重置订单编号
        if($rs) {
            error::_View($tpl,"订单取消成功",$backurl);
        }else {
            error::_View($tpl,"无法取消订单，请与管理员联系！",$backurl);

        }
    }
//查看订单方法
    function onshow() {
        $o      = &$this->o;
        $id     = GP('id');                        //订单 id
        $uinfo  = $this->ua->getuserByName($this->uname);
        $oinfo  = $o->getOrderInfo($id);           //指定 id 的订单详情
        $list   = $o->getOrderGoodsList($id);      //订单商品列表
        $tpl    = & $this -> loadview();
        $tpl -> assign("phone",$uinfo['U phone']);
        $tpl -> assign("address",$uinfo['U address']);
        $tpl -> assign("info",$oinfo);
        $tpl -> assign("list",$list);
        $tpl -> display('order_detail.html');      //加载查看订单页面模板
    }
}
?>
```

【代码解读】

订单管理控制器中定义了 4 种方法，分别是 onlist()、onsubmit()、oncancel()和 onshow()。这 4 个方法包括订单管理需要完成的 4 个“动作”。其中，onlist()被定义为获取订单列表、onsubmit()被定义为生成订单、oncancel()被定义为取消订单、onshow()被定义为获取指定订单的详情。

获取订单列表方法（默认加载动作），通过订单核心业务类（order）中的 getUserOrderList()方法获得该用户下的所有订单，再通过 tjUserOrderInfo()方法统计完成的订单情况，最后将状态和订单列表输出到视图层。

获得指定订单详情的方法 onshow()，通过传入指定订单号 getOrderInfo()获得订单详情，再匹配订单物品信息输出到指定的视图层。

说明：onsubmit()方法和 oncancel()方法都是处理订单状态的这里不再赘述。

2．模型

订单业务模型核心类（order.php）。该类中的方法按照完成的功能可以分为以下几类情况。

- 获得订单信息：获得订单相关信息的函数（方法）。
- 订单图片相关操作：对订单图片操作的函数（方法）。
- 订单管理操作：订单管理相关操作的函数（方法）。
- 获得用户订单情况：获得用户订单信息的函数（方法）。

❑ 单个订单的操作：单个订单的基本操作函数（方法）。

下面分别介绍这几种类型的函数集。首先介绍获得订单信息的函数集，该函数集被用来获得订单相关信息。代码如下：

```
<?
//是否正常读取
if(!defined('APPNAME')) {
    exit("没有权限");
}
uses("error");
class order extends model{
    /**
    * 函数名称： 获取订单编号
    * 输入参数： void
    * 返回参数： mixed
    **/
    function getOrderNewCode(){
        $db     = & $this->db();                //初始化数据库连接
        $query  = "select tp orderinfo seq.nextval from dual";
        $rt     = $db->Query($query);
        $arr    = $db->getrow($rt);
        $code   = $arr[0];                      //获得订单编号
        $today = getdate();                     //获得当前日期
        $code=$today[0];
        if($code) {                             //判断编号是否存在
            if(strlen($code)<6) {
                echo $code  = date('ymd').str_pad($code, 5, "0", STR_PAD_
                LEFT);
            }else {
                echo $code  = date('ymd').str pad(substr($code,0,8), 8,
                "0", STR PAD LEFT);
            }
            Return $code;                       //返回编号
        }else {
            Return false;
        }
    }

    /**
    * 函数名称： 获取订单状态
    * 输入参数：
        int $id     订单 id
        int $status 订单的状态
    * 返回参数： boolen
    **/
    function checkOrderStatus($id,$status){
        $db     = & $this->db();            //获得当前日期
        $id     = intval($id);              //id 合法性验证
        $query  = "select count(*) from tp_Orderinfo where o_id='$id' and
        o status='$status'";
        $rt     = $db->Query($query);
        $arr    = $db->getrow();
        if($arr) {
            if($arr[0]>0) {
                Return 1;
            }else {
```

```
                Return 0;                           //返回订单状态
            }
        }else {
            Return false;
        }
    }

    /**
    * 函数名称：  获得订单的详细信息
    * 输入参数：
        (int)$id:       用户 id
    * 返回参数：
        (array):操作成功返回记录数组，反之，返回 false
    **/
    function getOrderInfo($id){
        $db     = & $this->db("oracle_page");   //实例化数据库分页连接
        $uid    = intval($uid);
        $query  = "SELECT * FROM tp_Orderinfo where o_id='$id'";
        $rt     = $db->Query($query);           //返回查询结果
        $rtarr  = $db->getrow($rt);             //返回查询结果数组
        if($rtarr) {
            Return $rtarr;
        }else {
            Return false;
        }
    }
```

【代码解读】

获得订单信息的函数集被用来获得订单相关信息，一共包括 3 个方法，含义如下。

- getOrderNewCode()方法：作用是获取订单编号。
- checkOrderStatus($id,$status)：作用是获取订单状态。其中，参数$id 表示订单编号（id），参数$status 表示当前订单的状态。
- getOrderInfo()方法：作用是获得订单的详细信息。

订单图片相关操作的函数集，该函数集用来处理订单中有关图片的相关操作。该函数集代码如下：

```
/**
* 函数名称：  获得全部订单图片的总记录数
* 输入参数：
    (int)$id:       订单序号 id
* 返回参数：  int
**/
function getOrderGoodsCount($id){
    $db     = & $this->db("oracle page");   //实例化数据库分页连接
    $id     = intval($id);
    $query  = "SELECT count(*) FROM tp_Ordergoods where o_id='$id' ";
    $rt     = $db->Query($query);
    $rtarr  = $db->getrow($rt);             //获得数据数组
    if($rtarr) {
        Return $rtarr[0];                   //返回全部订单图片的总记录数
    }else {
        Return false;
    }
}
/**
```

```
* 函数名称：  获得全部订单图片的列表
* 输入参数：
    (int)$id:       订单序号 id
* 返回参数：
    (array):操作成功返回记录数组，反之，返回 false
**/
function getOrderGoodsList($id){
    $db     = & $this->db("oracle_page");  //实例化数据库分页连接
    $id = intval($id);
    $query  = "SELECT * FROM tp_ordergoods t1 ,tp_picinfo t2 where
    t1.O_id='$id' and t2.Picid = t1.picid
    $rt     = $db->Query($query);
    $rtarr  = $db->getRows($rt);           //获得数据数组
    if($rtarr) {
        Return $rtarr;                     //返回全部订单图片的数组
    }else {
        Return false;
    }
}

/**
* 函数名称：  删除订单中的图片记录
* 输入参数：
    mixed   $id     图片 id 或者图片 id 的数组
    int     $o_id   订单 id
* 返回参数：  boolen
**/
function delOrderGoods($id,$o id){
    $db     = & $this->db();               //实例化数据库连接
    $uid    = intval($uid);                //用户 id 有效性验证
    if(is_array($id)) {
        foreach($id as $key=>$val) {       //遍历用户 id 数组
            $val    = intval($val);
            $query  .= "delete FROM tp Ordergoods where picid='$val' and
            o id = '$o id'";
        }
    }else {
        $id     = intval($id);
        $query  = "delete FROM tp Ordergoods where picid='$id' and o id =
        '$o id'";
    }
    $rt     = $db->Query($query);          //执行查询命令
    Return $rt;
}
```

【代码解读】

获得订单图片相关操作的函数集被用来处理订单图片操作，一共包括 3 个方法，含义如下。

- getOrderGoodsCount($id)方法：作用是获得全部订单图片的总记录数。
- getOrderGoodsList($id)方法：作用是获得全部订单图片的列表。其中，参数$id 表示订单序号 id。
- delOrderGoods($id,$o_id)方法：作用是删除订单中的图片记录，其中，参数 id 表示图片 id 或者图片 id 的数组，参数$o_id 表示订单 id。

订单管理操作函数集，该函数集的作用是用来处理订单管理相关操作。该函数的代码如下：

```
/**
* 函数名称： 设定订单商品的价格
* 输入参数：
    int $id      订单 id
    strint  $price  订单商品的价格
* 返回参数： boolen
**/
function setOrderprice($id,$price){
    $db     = & $this->db();                   //实例化数据库连接
    $id     = intval($id);
    $query  =  "update  tp_Orderinfo  set  o_totalprice='$price'  where
    o_id='$id'";
    $rt     = $db->Query($query);              //执行操作命令
    Return $rt;                                //返回操作状态
}
/**
* 函数名称： 获得全部订单的列表
* 输入参数：
    (int)$status:   订单状态
    (int)$ord:      排序编号
    (int)$startnum: 读取记录的开始点
    (int)$endnum:   读取记录的结束点
* 返回参数：
    (array):操作成功返回记录数组，反之，返回 false
**/
function getOrderList($status,$ord,$startnum,$endnum){
    $db     = & $this->db("oracle_page");  //初始化数据库分页连接
    $query  = "SELECT t1.*,t2.u_truename FROM tp_Orderinfo t1 left join
    tp_users t2 on t1.u_id=t2.u_id";
    if($status!='') {                          //验证状态
        $query  .= " where t1.o_status='$status'";
    }
    $rt     = $db->PageQuery($query,$startnum,$endnum);
    $rtarr  = $db->getRows($rt);               //返回数据数组
    if($rtarr) {
        Return $rtarr;
    }else {
        Return false;
    }
}
/**
* 函数名称： 获得全部订单的总记录数
* 输入参数：
    (int)$status:   订单状态
* 返回参数： int
**/
function getOrderCount($status){
    $db     = & $this->db("oracle_page");  //初始化数据库分页连接
    $query  = "SELECT count(*) FROM tp_Orderinfo";
    if($status!='') {
        $query  .= " where o_status='$status'";
    }
    $rt     = $db->Query($query);              //执行 SQL
    $rtarr  = $db->getrow($rt);                //获得全部记录数
    if($rtarr) {
```

```
        Return $rtarr[0];                    //返回订单记录总数
    }else {
        Return false;
    }
}

/**
* 函数名称:  统计完成的订单情况
* 输入参数:
    (int)$uid:      用户 id
* 返回参数:
    (array):操作成功返回记录数组，反之，返回 false
**/
function tjUserOrderInfo($uid){
    $db     = & $this->db("oracle page");   //初始化数据库分页连接
    $query  = "SELECT count(*) num,sum(o_totalprice) totalprice FROM
    tp orderinfo where o status='1' and u id='$uid' ";
    $rt     = $db->Query($query);            //执行 SQL
    $rtarr  = $db->getrow($rt);
    if($rtarr) {
        Return $rtarr;                       //获得订单完成情况
    }else {
        Return false;
    }
}
```

【代码解读】

订单管理操作函数集被用来处理订单管理相关操作，一共包括 4 个方法，含义如下。

- setOrderprice($id,$price)方法：作用是设定订单商品的价格。其中，参数 id 表示订单 id，参数$price 表示订单商品的价格。
- getOrderList($status,$ord,$startnum,$endnum)方法：作用是获得全部订单的列表。其中，参数$status 表示订单状态，参数$ord 表示订单排序编号，参数$startnum 和$endnum 表示读取数据的开始位置和结束位置。
- getOrderCount($status)方法：作用是获得全部订单的总记录数。其中，参数$status 表示当前订单的状态。
- tjUserOrderInfo($uid)方法：作用是统计完成的订单情况。其中，参数 uid 表示用户的身份 id。

获得用户订单情况函数集，该函数集的作用是获得用户订单信息。该函数集的代码如下：

```
/**
* 函数名称:  获得用户全部订单的列表
* 输入参数:
    (int)$uid:      用户 id
    (int)$ord:      排序编号
    (int)$startnum: 读取记录的开始点
    (int)$endnum:   读取记录的结束点
* 返回参数:
    (array):操作成功返回记录数组，反之，返回 false
**/
function getUserOrderList($uid,$ord,$startnum,$endnum){
    $db     = & $this->db("oracle_page");        //实例化数据库分页连接
    $uid    = intval($uid);                      //id 有效性验证
    $query  = "SELECT * FROM tp_Orderinfo where U_id='$uid'";
```

```
    $rt     = $db->PageQuery($query,$startnum,$endnum);
    $rtarr  = $db->getrows($rt);
    if($rtarr) {
        Return $rtarr;                      //返回该用户的订单列表数组
    }else {
        Return false;
    }
}

/**
* 函数名称： 获得用户全部订单的总记录数
* 输入参数：
    (int)$uid:      用户 id
* 返回参数： int
**/
function getUserOrderCount($uid){
    $db     = & $this->db("oracle_page");   //实例化数据库分页连接
    $query  = "SELECT count(*) FROM tp_Orderinfo where u_id='$uid' ";
    $rt     = $db->Query($query);           //执行 SQL
    $rtarr  = $db->getrow($rt);
    if($rtarr) {
        Return $rtarr[0];                   //获得该用户全部订单总数数组
    }else {
        Return false;
    }
}
```

【代码解读】

以上是获得用户订单情况的函数（方法），该函数集的作用是获得用户订单信息，一共包括两个方法，含义如下。

- ❑ getUserOrderList($uid,$ord,$startnum,$endnum)方法：作用是获得用户全部订单的列表。其中，参数$uid 表示用户 id，参数$ord 排序编号，参数$startnum 和$endnum 表示读取数据的开始位置和结束位置。
- ❑ getUserOrderCount($uid)方法：作用是获得用户全部订单的总记录数。其中，参数$uid 表示用户 id。

单个订单的操作函数集，该函数集的作用是处理单个订单的基本操作。该函数集的代码如下：

```
    /**
    * 函数说明： 加入订单
    * 输入参数：
        mixed   $picid  加入订单的图片 id 或者图片 id 的数组
        int     $uid    用户 id
        string  $linkman    联系人
    * 返回参数： boolen
    **/
    function addOrder($linkman,$phone,$fax,$postcode,$address,$remark,
    $uid,$picid)    {                               //加入订单
        if(empty($picid)) Return false;
        $db     = & $this->db();                    //初始化数据库连接
        $uid    = intval($uid);                     //uid 校验
        $code   = $this->getOrderNewCode();         //初始化新订单
        $query  = "insert into tp_orderinfo
```

```
            (
            O id, O code, U id, O addtime, O totalprice, O status
            )
            values
            (
                '',
                '".$code."',
                ".$uid.",
                NOW(),
                '',
                '0'
            )";
        $rs     = $db->parse($query);             //执行操作命令（SQL 语句）
        $oid=mysql_insert_id();                   //获得操作 id 号
        if($rs) {
            if(is array($picid)) {
                //将订单信息循环插入数据库表中
                foreach($picid as $key=>$val) {
                    $val    = intval($val);
                    $query  = "insert into tp ordergoods
                    (
                        O id,picid
                    )
                    values
                    (
                        '".$oid."','".$val."'
                    )";
                    if(!$db->parse($query)) {   //执行操作命令（SQL 语句）
                        Return false;
                    }
                }
            }else {
                $picid  = intval($picid);         //图片 id 校验
                $query  = "insert into tp ordergoods
                    (
                        O id,picid
                    )
                    values
                    (
                        '".$oid."','".$picid."'
                    )";
                if(!$db->parse($query)) {
                    Return false;
                }
            }
        }else {
            Return false;
        }
        if($rs) {
            Return $code;                         //返回操作结果
        }else {
            Return false;
        }
    }
}

    /**
    * 函数说明：  编辑订单
    * 输入参数：
        int     $id         订单序号
```

```
    string  $linkman    联系人
    string  $phone      电话
    string  $fax        传真
    string  $postcode   邮编
    string  $address    地址
    string  $remark     备注
    string  $price      价格
* 返回参数： boolen
**/
function editOrder($id,$linkman,$phone,$fax,$postcode,$address,
$remark,$price){
    $db     = & $this->db();
    $query  = "update tp orderinfo set  o linkman='".$linkman."',
    o phone='".$phone."', o fax='".$fax."', o postcode='".$postcode."',
    o_address='".$address."', o_remark='".$remark."', o_totalprice='".
    $price."'where o id = '$id'";
    $rs     = $db->query($query);       //执行操作命令（SQL 语句）
    if($rs) {
        Return true;                    //编辑成功返回 true
    }else {
        Return false;
    }
}

/**
* 函数名称： 删除订单
* 输入参数：
    mixed   $id     订单 id
    int     $uid    用户 id
* 返回参数： boolen
**/
function delOrder($id,$uid){
    $db     = & $this->db();
    $id     = intval($id);              //订单 id 校验
    $uid    = intval($uid);             //用户 id 校验
    $query  = "delete FROM tp Orderinfo where o id='$id' and u id =
    '$uid'";
    $query  .= "delete FROM tp_Ordergoods where o_id='$id'";
    $rt     = $db->Query($query);       //执行操作命令（SQL 语句）
    Return $rt;                         //返回操作结果
}
```

【代码解读】

以上是单个订单的操作函数集的代码，该函数集的作用是处理单个订单的基本操作。一共包括 3 个基本方法，含义如下。

- addOrder($linkman,$phone,$fax,$postcode,$address,$remark,$uid,$picid)方法：作用是将图片商品加入订单。其中，参数表示图片商品的属性信息。
- editOrder($id,$linkman,$phone,$fax,$postcode,$address,$remark,$price)方法：作用是编辑指定的订单信息。其中，参数表示图片商品的属性信息。
- delOrder($id,$uid)方法：作用是删除指定的订单。其中，参数$id 表示订单 id 号，参数$uid 表示用户 id。

订单业务模型核心类程序负责处理全部与订单有关的业务逻辑，该类中涉及了众多的业务功能。但是，这些功能都是围绕着订单详情表（tp_orderinfo）和订单商品表

（tp_ordergoods）读写操作，例如获得用户全部订单的列表方法 getUserOrderList()，是根据用户 id 读取订单详情表（tp_orderinfo）的信息并生成列表数组。其他订单业务模型中的方法，都是相似的代码实现。

3．视图

在订单管理流程控制器中可以了解到，订单管理流程中涉及了 3 个视图展示，分别是订单列表视图 infor_order.html、订单详情视图 order_detail.html 和订单生成视图 shopping_step4.html。由于订单生成视图在前文已经介绍过了，这里重点介绍订单列表视图 infor_order.html 和订单详情视图 order_detail.html。

订单列表视图 infor_order.html，该视图将订单信息以列表的形式展示出来。设计部分重点是列表的循环显示，核心代码如下：

```
<table width="100%" border="0" cellpadding="1" cellspacing="1">
    <tr>
        <td height="30" colspan="6" align="right" bgcolor="#999999">
        <select name="ord" onchange="location='<%$orderurl%>ord/'+this.
        value+'/';">
            <option>选择排序方式 </option>
                      <option value="1">按下单时间排序</option>
                      <option value="3">按订单金额排序</option>
                      <option value="5">按订单状态排序</option>
         </select></td>
    </tr>
    <tr>
        <td width="14%" height="23" bgcolor="#CCCCCC" class="ti">订单号</td>
        <td width="22%" bgcolor="#CCCCCC" class="ti">收货单位</td>
        <td width="10%" align="center" bgcolor="#CCCCCC" class="ti">联系人
        </td>
        <td width="17%" align="center" bgcolor="#CCCCCC" class="ti">订单金
        额</td>
        <td width="14%" align="center" bgcolor="#CCCCCC" class="ti">订单状
        态</td>
        <td width="23%" align="center" bgcolor="#CCCCCC" class="ti">下单时
        间</td>
    </tr>
    <%section name=item loop=$list%>
    <tr>
        <td class="zt"><%$list[item].O code%></td>
        <td class="zt"><a href="?/c/order/act/show/id/<%$list[item].O
        id%>"><%$tname%></a></td>
        <td align="center" class="zt"><%$list[item].O linkman%></td>
        <td align="center" class="zt"><%$list[item].O_totalprice%></td>
        <td align="center" class="zt"><%if $list[item].O status==0%><a
            href="?/c/order/act/cancel/id/<%$list[item].O_id%>">取消</a>
            <%elseif
        $list[item].O_status==1%>已完成<%elseif
        $list[item].O status==11%>已作废<%/if%></td>
        <td align="center" class="zt"><%$list[item].O addtime%></td>
    </tr>
    <tr>
        <td colspan="6" class="zt">
        <hr width="100%" size="1" />
        </td>
    </tr>
```

```
    <%/section%>
    <tr>
        <td class="zt"> </td>
        <td class="zt"> </td>
        <td align="center" class="zt"> </td>
        <td class="zt"> </td>
        <td align="center" class="zt"> </td>
        <td align="center" class="zt"> </td>
    </tr>
    <form method="post" action="<%$pagebar.linkparam%>">
    <tr>
        <td colspan="6" align="right" class="zt"><span class="ti">共
        [<%$pagebar.totalpages%>]页
        <a href="<%$pagebar.linkparam%>page/<%$pagebar.prevpage%>">上一页
        </a> <a
            href="<%$pagebar.linkparam%>page/<%$pagebar.nextpage%>">下一
            页</a> 转到 <input
            name="page" type="text" size="3" /> 页 <input type="submit"
            name="Submit" value="确认" /> </span></td>
    </tr>
    </form>
    <tr>
        <td colspan="6" class="zt"> </td>
    </tr>
    <tr>
        <td colspan="6" align="right" class="zt">截至<%$currentdate%>，您
        共有<%$completeorder.NUM%>张订单完成交易，累计消费<%$completeorder.
        TOTALPRICE|default:0%>元</td>
    </tr>
</table>
```

在浏览器中输入地址：

```
http://localhost/pt/?/c/order/
```

效果如图 20.18 所示。从图中可以看到，订单列表界面中显示了全部已经生成的订单，其中还可以查看订单的状态等详细信息。

订单管理

选择排序方式

订单号	收货单位	联系人	订单金额	订单状态	下单时间
1288342497	清华大学出版社			取消	2010-10-29 16:54:57
1288342448	清华大学出版社			取消	2010-10-29 16:54:08
1288342644	清华大学出版社			取消	2010-10-29 16:57:24
1288342444	清华大学出版社			已作废	2010-10-29 16:54:04
1288342398	清华大学出版社			取消	2010-10-29 16:53:18
1288342365	清华大学出版社			取消	2010-10-29 16:52:45
1288342361	清华大学出版社			取消	2010-10-29 16:52:41
1287655443	清华大学出版社			已作废	2010-10-21 18:04:03

共[1]页 上一页 下一页 转到 页 确认

图 20.18　订单管理列表

订单详情视图 order_detail.html，该视图的作用是显示指定订单的详情。核心代码如下：

```
<table cellspacing="0" cellpadding="0" width="95%" align="center"border="0">
<tbody>
    <tr>
        <td width="100%">
        <table bordercolor="#000000" cellspacing="0" cellpadding="3"
            width="100%" border="1">
            <tbody>
                <tr>
                    <td width="27%" class="ti">订单号： <%$info.O_code%>
                    </td>
                    <td width="36%" class="ti">收货人： <%$info.O_linkman%>
                    </td>
                    <td width="37%" class="ti">联系电话： <%$phone%></td>
                </tr>
                <tr>
                    <td colspan="3" class="ti"><%$address%></td>
                </tr>
            </tbody>
        </table>
        </td>
    </tr>
    <tr>
        <td bgcolor="#f5f5cc">
        <table bordercolor="#000000" cellspacing="0" cellpadding="3"
            width="100%" border="1">
            <tbody>
                <tr bgcolor="#0066cc">
                    <td width="11%" align="middle" class="news"><font
                        color="white">编号</font></td>
                    <td width="14%" align="middle" class="news"><font
                        color="white">作品号</font></td>
                    <td width="63%" align="middle" class="news"><font
                        color="white">作品名称</font></td>
                    <td width="12%" align="middle" class="news"><font
                        color="white">作者</font></td>
                </tr>
                <%section name=item loop=$list%>
                <tr>
                    <td align="middle" class="ti"><%$smarty.section.item.
                    index next%></td>
                    <td align="left" class="ti"><%$list[item].Code%></td>
                    <td align="left" class="ti"><%$list[item].Name%></td>
                    <td align="middle" class="ti"><%$list[item].artist%>
                    </td>
                </tr>
                <%/section%>
            </tbody>
        </table>
        </td>
    </tr>
    <tr>
        <td bgcolor="lightblue">
        <table bordercolor="#000000" cellspacing="0" cellpadding="3"
            width="100%" border="1">
            <tbody>
                <tr>
```

```
                    <td class="ti"> </td>
                    <td width="35%" class="ti">订购总额：<b>
                    <%$info.O_totalprice%> 元</b></td>
                </tr>
            </tbody>
        </table>
        </td>
    </tr>
    <tr>
        <td>
        <table bordercolor="#000000" cellspacing="0" cellpadding="3"
            width="100%" border="1">
            <tbody>
                <tr>
                    <td width="50%" class="ti">订购日期： <%$info.O_
                    addtime%></td>
                    <td width="50%" class="ti">订单完成日期： <%$info.O_
                    optiontime%></td>
                </tr>
            </tbody>
        </table>
        </td>
    </tr>
    <tr>
        <td>
        <table bordercolor="#000000" cellspacing="0" cellpadding="3"
            width="100%" border="1">
        </table>
        </td>
    </tr>
</tbody>
</table>
```

在订单列表中点击收货单位将会显示指定订单的详情，效果如图 20.19 所示。从图中可以看到订单的详细信息。

定单详细信息

订单号：1288342497	收货人：		联系电话：010-12345
北京市海淀区知春路			
编号	作品号	作品名称	作者
1	pic03	测试名称3	
		订购总额：元	
订购日期：2010-10-29 16:54:57		订单完成日期：	

打印此订单　浏览合同副本　打印合同副本

合同副本以PDF方式浏览 下载浏览器

图 20.19　订单详情

20.6.3　收藏夹管理

对于收藏夹大家并不陌生，例如浏览器中的收藏夹功能，它用来储存网址，方便使用者通过提示列表直接找到站点的域名。在图片交易平台的收藏夹管理也实现了相似的功能，只是保存和管理的内容不是网站域名和链接，而是用户放入收藏夹中的商品。

下面介绍收藏夹管理部分的控制器设计。创建控制器文件（c_collection.php），该控制器将控制收藏管理相关的流程。

1．控制器

收藏夹管理控制器（c_collection.php）的代码如下：

```
<?
//是否正常读取
if(!defined('APPNAME')) {                              //权限验证
    exit("没有权限");
}
uses("error");
class c_collection extends controller {
    function c_collection() {
        loadModel("user_authz");                       //加载用户验证类
        $this->ua    = & new user_authz;               //实例化用户验证类
        if($this->ua->isLogin()) {
            $this->uid  = $this->ua->getAuthzId();//获得 uid
            $this->uname= $this->ua->getAuthzName();
        }else{
            $tpl    = & $this -> loadview();
            $bu     = "?/c/login/";
            error:: view($tpl,'没有访问权限！',$bu);
        }
        loadModel("collection");                       //加载收藏业务模型
        $this->co = & new collection;                  //实例化收藏夹业务模型
    }
//默认加载动作
    function onMain() {
        if(GP('act')=="") {
            $this -> onlist();                         //默认加载 onlist 方法
        }
    }
//收藏夹列表方法
    function onlist($message='') {
        uses("pagebar");                               //加载分页基类
        $co      = &$this->co;
        $uinfo   = $this->ua->getuserByName($this->uname);
        $pb      = new pagebar;
        $pz      = 10;                                 //页面跨度
        $lp      = url('collection','list',array());
        $tr      = $co->getCollectionCount($this->uid);
                                                       //计算收藏夹中图片的个数
        $pbar    = $pb-> execute($tr,$pz,$lp);
        $pbar['startrow']   = 0;                       //起始记录
        $pbar['endrow']=10;                            //跨度
        $list    = $co->getCollectionList($this->uid,$pbar['startrow'],
        $pbar['endrow']);
        $tpl     = & $this -> loadview();
        $tpl -> assign("pagebar",$pbar);
        $tpl -> assign("Tname",$uinfo['U truename']);
        $tpl -> assign("message",$message);
        $tpl -> assign("list",$list);
        $tpl -> display('infor_coll.html');
    }
//加入收藏夹方法
```

```
    function onadd(){
        $this->debugon();
        $co     = &$this->co;
        $backurl= getBackUrl();                        //返回链接
        $picid  = GP('id');
        $rs     = $co -> addcollection($picid,$this->uid);
                                                       //加入收藏夹模型方法
        $tpl    = & $this -> loadview();
        if($rs) {
            error::_View($tpl,"成功加入收藏夹",$backurl);
        }else {
            error::_View($tpl,"无法加入收藏夹，请与管理员联系！",$backurl);

        }
    }
//从收藏夹中移除方法
    function ondel() {
        $co     = &$this->co;
        $backurl= getBackUrl();                        //返回链接
        $picid  = GP('id');                            //图片编号
        $rs     = $co -> delcollection($picid,$this->uid); //移除操作
        $tpl    = & $this -> loadview();
        if($rs) {
            $this->onlist("成功从收藏夹中删除！");
        }else {
            error:: View($tpl,"无法从收藏夹中删除，请与管理员联系！",$backurl);

        }
    }
}
?>
```

【代码解读】

默认加载动作 onlist()方法，该方法被定义为显示当前用户收藏的全部商品。具体的实现步骤是首先根据 uid 通过 getuserByName()方法获得该用户的全部信息，然后调用收藏业务模型类中的 getCollectionList()方法。该方法被定义为通过指定 uid 获得列表数组，形式如下：

```
$list = $co->getCollectionList($this->uid,$pbar['startrow'],$pbar['endrow']);
```

将物品加入和移除收藏夹的流程比较简单，核心是用到了收藏业务模型类中的 addcollection()方法（加入）和 delcollection()方法（移除）。代码形式如下：

```
//加入收藏夹
$rs = $co -> addcollection($picid,$this->uid);
//将指定物品移除收藏夹
$rs = $co -> delcollection($picid,$this->uid);
```

2. 模型

收藏夹管理的控制器部分被设计用来控制收藏管理的流程，其中，业务功能交给模型层的类来具体实现。下面介绍收藏夹管理中涉及到的业务模型中的方法，相应的代码如下：

```
<?php
//是否正常读取
```

```
if(!defined('APPNAME')) {                              //权限验证
    exit("没有权限");
}
uses("error");
class collection extends model{

    /**
    * 函数名称： 获得全部收藏夹图片的列表
    * 输入参数：
        (int)$uid:      用户 id
        (int)$startnum: 读取记录的开始点
        (int)$endnum:   读取记录的结束点
    * 返回参数：
        (array):操作成功返回记录数组，反之，返回 false
    **/
    function getCollectionList($uid,$startnum,$endnum){
        $db     = & $this->db("oracle_page");  //实例化数据库
        $uid    = intval($uid);
        $query  = "SELECT t1.id,t1.adddate,t2.* FROM tp_collection t1 ,
        tp picinfo t2 where t1.Uid='$uid' and t2.Picid = t1.Picid ";
                                                       //执行 SQL
        $rt     = $db->PageQuery($query,$startnum,$endnum);
        $rtarr  = $db->getRows($rt);           //获得全部收藏夹图片的列表数据集
        if($rtarr) {
            Return $rtarr;                     //返回全部收藏夹图片的列表数组
        }else {
            Return false;
        }
    }

    /**
    * 函数名称： 获得全部收藏夹图片的总记录数
    * 输入参数：
        (int)$uid:      用户 id
    * 返回参数： int
    **/
    function getCollectionCount($uid){
        $db     = & $this->db("oracle page");  //实例化数据库
        $query  = "SELECT count(*) FROM tp_collection where Uid='$uid' ";
                                               //执行 SQL
        $rt     = $db->Query($query);          //执行 SQL
        $rtarr  = $db->getrow($rt);            //获得全部收藏夹图片的总记录数
        if($rtarr) {
            Return $rtarr[0];
        }else {
            Return false;
        }
    }

    /**
    * 函数名称： 删除收藏夹的图片记录
    * 输入参数：
        mixed   $id     收藏夹 id 或者收藏夹 id 的数组
        int     $uid    用户 id
    * 返回参数： boolen
    **/
    function delcollection($id,$uid){
```

```
    $db     = & $this->db();              //实例化数据库
    $uid    = intval($uid);
    if(is array($id)) {
        $val    = implode(',',$id);      //数组的内容组合成一个 id 字符串
        $query  = "delete FROM tp collection where id in ($val) and Uid
        = '$uid'";
    }else {
        $id     = intval($id);            //图片 id
        $query  = "delete FROM tp collection where id='$id' and Uid =
        '$uid'";
    }
    $rt     = $db->Query($query);
    Return $rt;
}

/**
* 函数名称：  收藏夹中图片是否已经存在
* 输入参数：
    string  $picid  图片 id
    int     $uid    用户 id
* 返回参数：  boolen
**/
function is exists($picid,$uid){
    $db     = & $this->db();              //实例化数据库
    $uid    = intval($uid);
    $query  = "select count(*) FROM tp_collection where Uid = '$uid' and
    Picid='$picid'";
    $rt     = $db->Query($query);        //执行 SQL
    $rs     = $db->getrow($rt);          //返回执行结果
    if($rs) {
        if($rs[0]>0) {
            Return true;
        }else {
            Return false;
        }
    }else {
        Return false;
    }
}

/**
* 函数说明：  加入收藏夹
* 输入参数：
    mixed   $picid  加入收藏夹的图片 id 或者图片 id 的数组
    int     $uid    用户 id
* 返回参数：  boolen
**/
function addcollection($picid,$uid){        //加入
    $db     = & $this->db();
    $uid    = intval($uid);
    $picid  = intval($picid);
    //加入收藏夹的图片 id 或者图片 id 的数组
    if(!$this->is exists($picid,$uid)) {
        $query  = "insert into tp collection(id,Uid,Picid,adddate)
        values('','".$uid."','".$picid."',NOW())";
    }
    $rt = $db->Query($query);
    Return $rt;
```

```
    }
}
?>
```

【代码解读】

收藏夹业务模型的核心任务是对"用户收藏夹"tp_collection 这张表中的数据进行管理，获得全部收藏夹图片的列表的方法 getCollectionList()。该方法通过用户 id（即 uid）匹配获得用户资料信息，然后通过图片 id（即 picid）匹配出图片商品的详情，再用数组的方式输出。

加入收藏夹和从收藏夹中移除图片，也是通过对"用户收藏夹"tp_collection 表中进行插入和删除操作，来实现业务逻辑的。

3．视图

收藏夹管理部分只涉及一个视图文件（infor_coll.html），该视图文件的功能是将该用户的收藏商品信息以列表形式显示出来。列表部分的代码如下：

```
<table width="100%" border="0" cellpadding="1" cellspacing="1">
    <tr>
        <td height="30" colspan="6" bgcolor="#999999" class="news">
        <%$Tname%>
        共收藏<%$pagebar.totalrows%>方图片</td>
    </tr>
    <form action="" method="post" name="form1" id="form1">
    <tr>
        <td width="6%" height="23" align="center" bgcolor="#CCCCCC"
        class="ti">选中
        </td>
        <td width="16%" align="center" bgcolor="#CCCCCC" class="ti">作品号
        </td>
        <td width="37%" align="center" bgcolor="#CCCCCC" class="ti">作品名
        称</td>
        <td width="17%" align="center" bgcolor="#CCCCCC" class="ti">作者
        </td>
        <td colspan="2" align="center" bgcolor="#CCCCCC" class="ti">放入时
        间</td>
    </tr>
    <%section name=item loop=$list%>
    <tr>
        <td align="center" class="zt"><input type="checkbox" name="id[]"
            value="<%$list[item].id%>"
        id="selid<%$smarty.section.item.index%>" /></td>
        <td align="center" class="zt"><%$list[item].Code%></td>
        <td align="center" class="zt"><%$list[item].Name%></td>
        <td align="center" class="zt"><%$list[item].Artist%></td>
        <td colspan="2" align="center" class="zt"><%$list[item].adddate%>
        </td>
    </tr>
    <tr align="center">
        <td colspan="6" class="zt">
        <hr width="100%" size="1" />
        <input type="hidden" name="picid[]" value="<%$list[item]. Picid%>"/>
        <input type="hidden" name="name[]" value="<%$list[item].Name%>"/>
        <input type="hidden" name="artist[]" value="<%$list[item]. Artist%>"/>
        <input type="hidden" name="psize[]" value="<%$list[item].Psize%>"/>
        </td>
```

```
    </tr>
    <%/section%></form>
    <tr>
        <td height="5" colspan="6"></td>
    </tr>
    <form method="post" action="<%$pagebar.linkparam%>">
    <tr>
        <td colspan="6" align="right" class="zt"><span class="ti">共
        [<%$pagebar.totalpages%>]页
    <a href="<%$pagebar.linkparam%>page/<%$pagebar.prevpage%>">上一页</a>
    <a href="<%$pagebar.linkparam%>page/<%$pagebar.nextpage%>">下一页</a>
    转到<input name="page" type="text" size="3" />页
    <input type="submit" name="Submit" value="确认" /> </span></td>
    </tr>
    </form>
    <tr>
        <td height="5" colspan="6"></td>
    </tr>
    <tr>
        <td colspan="6" class="zt">
        <table width="100%" border="0" cellpadding="0" cellspacing="0">
            <tr>
                <td width="11%" class="zt 2"><a href="#"
                    onclick="selall();return false;">全部选中</a></td>
                <td width="58%" class="zt 2"><a href="#"
                    onclick="selnone();return false;">取消选中</a></td>
                <td width="16%" class="zt 2"><a href="#"
                    onclick="buy();return false;">购买选中的作品</a></td>
                <td width="15%" class="zt_2"><a href="#"
                    onclick="del();return false;">删除选中的作品</a></td>
            </tr>
        </table>
        </td>
    </tr>
</table>
```

在用户管理中心中选择收藏夹管理选项，收藏夹管理列表预览效果如图 20.20 所示。从图中可以看到已经放入收藏夹中的图片商品信息。



图 20.20　收藏夹管理列表

当用户勾选收藏夹列表中的商品图片后，单击“删除选中的作品”链接，效果如图 20.21 所示。从图中可以看到提示删除收藏夹中图片商品的提示信息。



图 20.21　删除选中作品

20.6.4　出版社信息管理

图片交易平台的用户是出版社和机构，订购的流程也是针对出版社和机构用户。由于出版社的联系人和联系方式有时会有变化，因此提供用户修改基本信息和账号信息的管理功能，以保障订单的准确和账号的安全。

出版社信息在图片交易平台中被定义为“用户信息”。因此，出版社信息管理的控制器部分被放在用户业务控制器（c_user.php）。

1. 控制器

出版社信息管理控制器代码设计如下：

```
<?
//是否正常读取
if(!defined('APPNAME')) {                              //权限认证
    exit("没有权限");
}
uses("error");
class c user extends controller {
//构造函数
    function c user() {
        loadModel("user_authz");                       //加载用户认证模型
        $this->ua   = & new user authz;                //实例化用户认证类
        if($this->ua->isLogin()) {
            $this->uid  = $this->ua->getAuthzId(); //登录后获得 uid
            $this->uname= $this->ua->getAuthzName();
        }else{
            $tpl    = & $this -> loadview();
            $bu     = "?/c/login/";
            error:: view($tpl,'没有访问权限！',$bu);
        }
        loadModel("user");                             //加载用户模型类
        $this->u = & new user;
    }
    //默认动作
    function onMain() {
        if(GP('act')=="") {
            $this -> onshowuser();                    //默认加载 onshowuser 方法
```

```
        }
    }
//用户信息详情方法
    function onshowuser($mess='') {
        $u      = &$this->u;
        $id     = $this->uid;                //获得用户 uid
        $uinfo  = $u -> getUserById($id);    //获得指定 id 的用户详情
        $tpl    = & $this -> loadview();
        $tpl -> assign("message",$mess);
        $tpl -> assign("info",$uinfo);
        $tpl -> display('infor basic.html');
    }
    //编辑用户方法
    function onedit() {
        $u      = &$this->u;
        $ua     = &$this->ua;
        $id     = $this->uid;
        $name   = $this->uname;
        $parr   = GP();
        //检查
        if(empty($parr['address'])) {       //检查地址不为空
            error:: Set("地址不能为空！");
        }
        if(empty($parr['phone'])) {         //检查电话不为空
            error::_Set("电话不能为空！");
        }
        if(empty($parr['linkman'])) {       //检查联系人不为空
            error::_Set("联系人不能为空！");
        }
        $tpl    = & $this -> loadview();
        if(error::_IS()) {
            error:: View($tpl,"",url("user",""));
        }else{
    //编辑修改
    if($u->editUser($id,$parr['address'],$parr['postcode'],$parr['phone'
    ], $parr['fax'],$parr['linkman'],$parr['email'])) {
                $this->onshowuser("编辑修改成功");
            }else {
                error:: View($tpl,"无法编辑资料,请与管理员联系!",url ("user",""));
            }
        }
    }
    //修改密码方法
    function onchangepwd() {
        $u      = &$this->u;
        $ua     = &$this->ua;
        $id     = $this->uid;
        $name   = $this->uname;
        $parr   = GP();
        //密码检查
        if(empty($parr['oldpwd'])||empty($parr['pwd'])||empty($parr
        ['repwd'])) {
            error::_Set("密码不能为空！");
        }
        if( $this->wrongpwd($parr['pwd'])||$this->wrongpwd($parr['repwd'])) {
            error::_Set("密码必须是由大小写英文字母及数字组成的 6-12 位字串！");
```

```
        }
        if($parr['pwd']!=$parr['repwd']) {
            error:: Set("两次填写的新密码不一致！");
        }
        $uinfo  = $ua -> getUserByName($name);
        if($uinfo['U_PASSWORD']!=md5($parr['oldpwd'])) {
            error:: Set("旧密码错误！");
        }
        $tpl    = & $this -> loadview();
        if(error:: IS()) {
            error:: View($tpl,"",url("user",""));
        }else{
            if($u->changepwd($id,$parr['pwd'])) {
                $this->onshowuser("密码修改成功");
            }else {
                error:: View($tpl,"无法修改密码，请与管理员联系！",url("user",""));
            }
        }
    }
}
?>
```

【代码解读】

出版社信息管理控制器包含两个控制流程：

（1）出版社基本信息修改（地址、联系人等）。

（2）用户登录密码的修改。

围绕着这两个流程控制器部分，涉及 3 个方法，即显示“用户信息”方法 onshowuser()、编辑“用户信息”方法 onedit()和修改密码方法（密码重置）onchangepwd()。代码实现部分首先实例化用户业务核心类，使用 editUser()方法编辑用户基本信息，使用 changepwd()方法重置用户密码。

2. 模型

出版社信息管理部分的模型属于“用户业务”核心类，在控制器部分用到几个业务方法来实现用户信息修改、用户密码重置等业务逻辑。下面具体介绍这几个方法，代码如下：

```
/**
* 函数说明： 编辑用户
* 输入参数:
* 返回参数： boolen
**/
function editUser($id,$address,$postcode,$phone,$fax,$linkman,$email){ //
加入
    $db     = & $this->db();
    //编辑用户 SQL 更新操作
    $query = "update tp_users set u_address='".$address."',
    u_postcode='".$postcode."', u_phone='".$phone."', u_fax='".$fax."',
    u_linkman='".$linkman."', u_email='".$email."'  where u_id='".$id."'";
    $rt = $db->Query($query);
    Return $rt;
}
/**
* 函数说明： 密码重置
* 输入参数:
    int $id
```

```
    string  $pwd
* 返回参数:  boolen
**/
function changepwd($id,$pwd){ //加入
    $db     = & $this->db();
    //密码重置 SQL 更新操作
    $query = "update tp users set u password='".md5($pwd)."' where u id='
    ".$id."'";
    $rt = $db->Query($query);
    Return $rt;
}
```

【代码解读】

出版社管理业务模型，实现业务逻辑的实质是对会员资料表 tp_users 中字段信息的重置。以用户 id（uid）为条件调用用户编辑 onedit()方法，更新指定字段的用户资料信息，在控制器文件中，用户 id（uid）通过实例化会员业务核心类（user_authz.php）中的方法获得。

代码如下：

```
$this->uid  = $this->ua->getAuthzId();
```

3. 视图

出版社信息管理部分的视图，分为出版社基础信息修改和出版社登录密码修改两个部分。为了使用方便，将两个部分放在同一个视图文件中显示。

先来看出版社基础信息修改的部分编码，代码如下：

```
<form action="?/c/user/act/edit" method="post">
<table width="100%" border="0" cellpadding="0" cellspacing="0">
    <tr>
        <td width="36%" height="30" align="right" class="news2">出版社名称:
        </td>
        <td width="2%" class="news2"> </td>
        <td width="62%" class="zt"><%$info.u truename%></td>
    </tr>
    <tr>
        <td height="30" align="right" class="news2">地址: </td>
        <td class="news2"> </td>
        <td class="news2"><input name="address" type="text" id="address"
            value="<%$info.u address %>" size="40" /></td>
    </tr>
    <tr>
        <td height="30" align="right" class="news2">邮编: </td>
        <td class="news2"> </td>
        <td class="news2"><input name="postcode" type="text"
            id="postcode" value="<%$info.u postcode%>" /></td>
    </tr>
    <tr>
        <td height="30" align="right" class="news2">电话: </td>
        <td class="news2"> </td>
        <td class="news2"><input name="phone" type="text" id="phone"
            value="<%$info.u phone%>" /></td>
    </tr>
```

```
    <tr>
        <td height="30" align="right" class="news2">传真：</td>
        <td class="news2"> </td>
        <td class="news2"><input name="fax" type="text" id="fax" value=
        "<%$info.u Fax%>"
        /></td>
    </tr>
    <tr>
        <td height="30" align="right" class="news2">联系人：</td>
        <td class="news2"> </td>
        <td class="news2"><input name="linkman" type="text" id="linkman"
            value="<%$info.u linkman%>" size="10" /></td>
    </tr>
    <tr>
        <td height="30" align="right" class="news2">电子邮件 ：</td>
        <td class="news2"> </td>
        <td class="news2"><input name="email" type="text" id="email"
            value="<%$info.u email %>" size="30" /></td>
    </tr>
    <tr>
        <td colspan="3" align="center"><input name="id" type="hidden"
            id="id" value="<%$info.u id%>" size="30" /> <input
            type="image" src="<%$image dir%>think.gif" width="115"
        height="45" /></td>
    </tr>
</table>
</form>
```

在浏览器中预览出版社基础信息修改页面，效果如图 20.22 所示。从图中可以看到出版社的基本信息，重新输入信息后单击“确认”按钮完成修改工作。

图 20.22　出版社基础信息修改

出版社登录密码修改的视图是常见的展示方式，用户需要输入旧的密码，然后输入两次新设定的密码确认修改。

在浏览器中预览出版社登录密码修改的页面效果，如图 20.23 所示。

图 20.23　出版社登录密码修改

20.7　小　　结

图片交易平台模块是一个典型的平台程序的应用，本章学习的重点是如何使用 MVC 的代码组织形式（框架）实现平台管理类的程序实现。

本章首先从图片交易平台的系统概述讲起，这部分包括基础架构设计和核心功能描述，属于需求分析的阶段，使读者明确需求。第 20.2 节介绍 MVC 实现原理。首先介绍 MVC 基础，然后具体地讲解了 MVC 的 3 个组成构件，即 Model（模型）、View（视图）、Controller（控制器）。第 20.3 节介绍数据库设计。首先确认需求分析，然后介绍数据库 E-R 关系图，最后是数据表的设计。从第 20.4 节开始进入到具体图片交易业务流程开发中，图片搜索及展示是最基础的业务流程。这个流程包括 3 个部分，即图片列表、图片搜索和图片详情。第 20.5 节介绍图片收藏及订购，这个是图片交易平台的核心业务流程，它包括图片收藏、图片购物车和订单处理流程。读者在掌握 MVC 的组织形势后，需要注意对具体业务流程的理解。第 20.6 节介绍用户管理中心是用户管理基本资料和处理订购等业务流程的功能模块。首先介绍用户登录验证的代码实现，这部分也是一个通用的模块，涉及用户私有操作的部分都使用该模块处理身份认证。然后介绍了订单管理、收藏夹管理和出版社信息管理，这 3 个部分囊括了用户在业务使用中的全部操作管理。

第21章　在线购物平台(PHP+jQuery+B2C)

在线购物就是通过互联网检索商品信息，并通过电子订购单发出购物请求，然后进行网上支付、下线送货完成在线购物的行为。互联网的普及和购物系统安全性能的提升，大大刺激了网络交易量的增长。根据相关统计结果显示，在线购物逐渐成为人们的基本消费行为之一。

本章将以在线购物为重点，介绍其中涉及的产品管理、订单管理、会员管理、购物车等相关内容。读者通过本章的学习，可以了解到在线购物的基本原理、设计规划；在业务模型层面，可以了解到在线购物涉及的业务模型的代码实现，以及如何设计扩展已有的应用。

本章主要涉及的知识点如下。

- 企业与消费者电子商务（B2C，Business To Customer）：即企业（商业机构）借助互联网开展在线销售活动，服务对象是最终消费者。
- Web 平台的体系结构：客户端表现层、应用服务层和数据核心层。
- jQuery：是一个快速、简洁的 JavaScript 库，使用户能更方便地处理 HTML documents、events、实现动画效果，并且方便地为网站提供 AJAX 交互。
- ImageDestroy ();函数：该函数的作用是销毁图像，并释放与其关联的内存（在创建图像后需要释放内存完成本次操作，避免数据被两次污染）。

21.1　系统概述

在线购物平台模块属于电子商务的一个具体应用，因此在介绍在线购物平台之前先介绍电子商务的含义和电子商务的几种类型。

目前，国际上对电子商务尚无统一的定义。通常提到的电子商务概念，狭义上是特指在互联网上进行的交易。广义上，指的是企业利用现代化信息技术开展的一切商务活动，它既包括网上交易，也包括企业内部及企业之间的协作与协调。

按照参与对象的不同可以分为如下两种。

（1）企业间电子商务（B2B，Business To Business）：企业与企业之间通过互联网进行产品、服务及信息的交换。

（2）企业与消费者电子商务（B2C，Business To Customer）：企业（商业机构）借助互联网开展在线销售活动，服务对象是最终消费者。

B2C 模式也是目前互联网领域最常见的、与普通消费者关系最密切的服务模式，本章介绍的在线购物平台模块就属于 B2C 模式的典型应用。

21.1.1　流程概述

前面介绍过，在线购物平台属于电子商务领域中 B2C 模式的应用，因此该系统的设计就是围绕着企业（商业机构）、消费者与交易平台的交互进行的。

首先介绍下电子商务系统 Web 平台的开发方式。Web 平台的体系结构分为以下 3 层。

（1）客户端表现层（客户端程序）：该层负责与用户相关的界面交互表现。

（2）应用服务层（服务端）：该层又可以细分为 Web 服务输出和应用服务处理。其中，Web 服务输出负责处理客户端提交过来的请求；应用服务处理则负责管理相关业务对应的应用逻辑，即所谓的事务处理。

（3）数据核心层：该层由数据库程序构成，主要负责交易过程中的数据处理操作（存储、备份、同步等）。

具体各层之间的逻辑关系如图 21.1 所示。

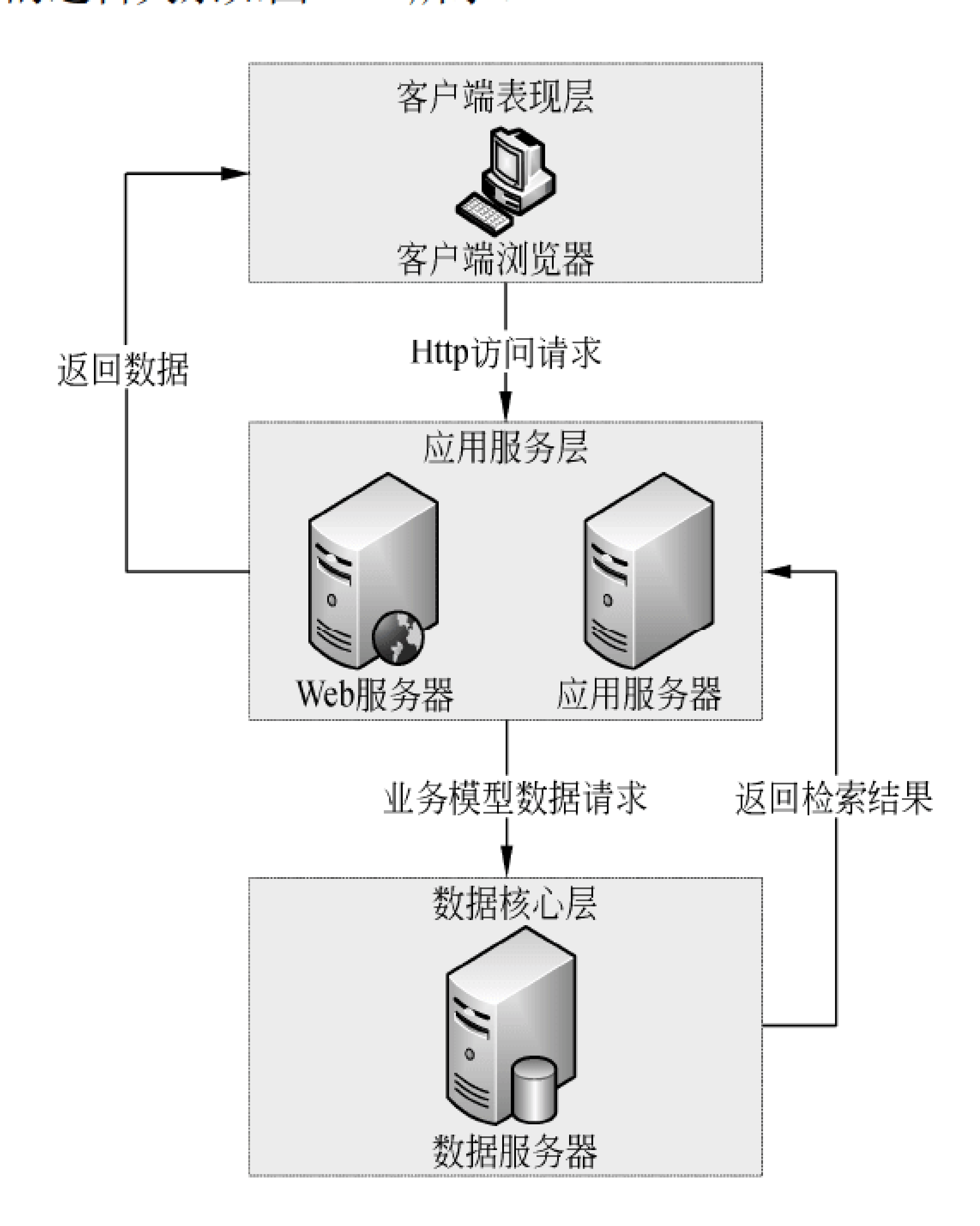


图 21.1　电子商务系统 Web 平台体系架构

在线购物平台模块属于电子商务 B2C 模式的具体应用，因此在了解了电子商务系统 Web 平台的体系架构后，介绍在线购物平台的业务流程。

注意：在线购物平台属于电子商务 B2C 模式，因此具体的业务流程也要遵循电子商务系统的基本架构，是从属的关系。

下面介绍在线购物的一般流程，因为具体的需求差异在局部可以略有调整，这里只介绍标准模式，读者可以结合具体的业务需求自行调整。在线购物的标准流程，如图 21.2 所示。

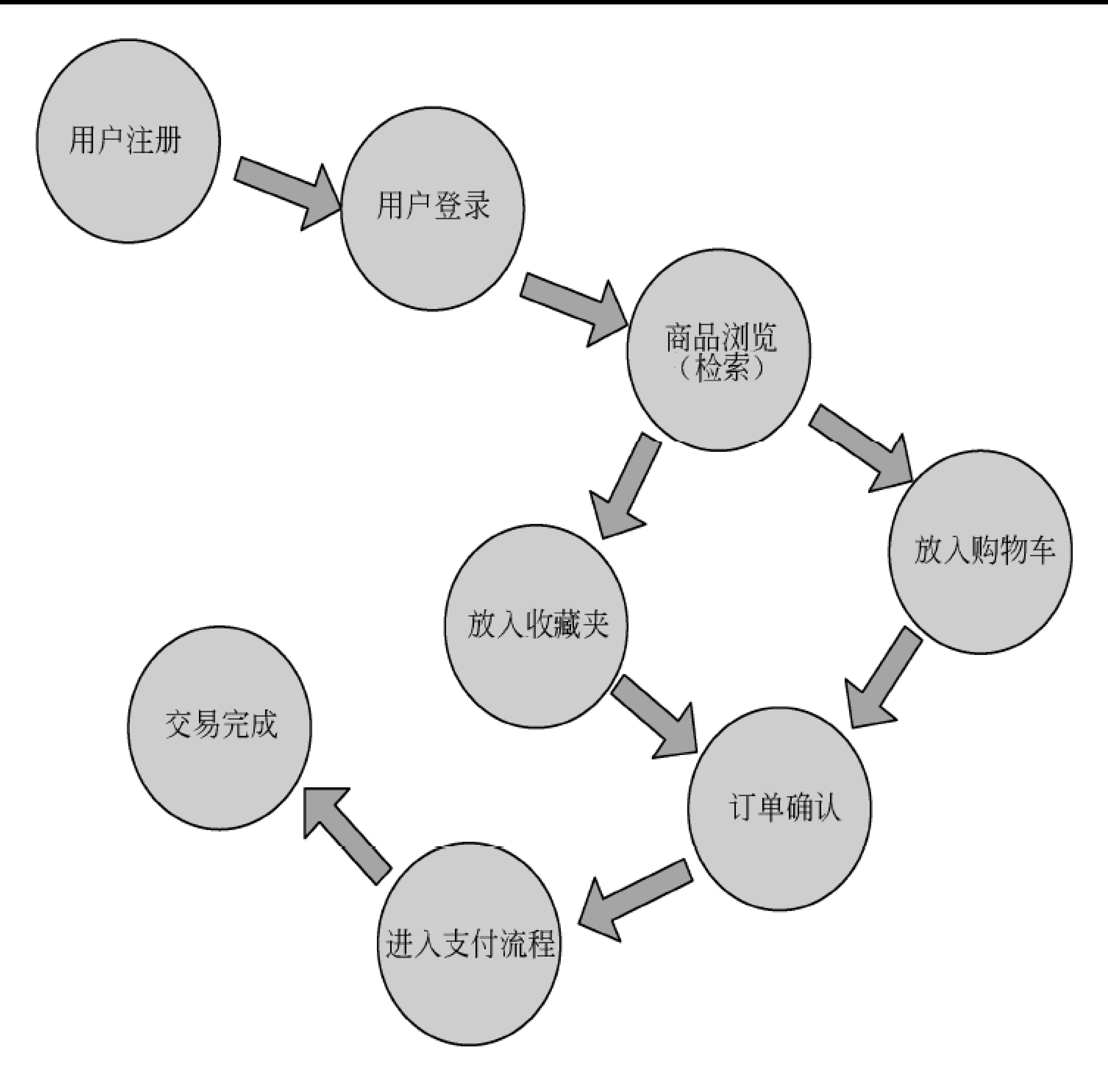


图 21.2　在线购物标准流程图

21.1.2　功能说明

通过前文的介绍，读者可以了解到在线购物网站主要功能包括：

（1）产品管理。

（2）订单管理。

（3）会员管理等模块。

（4）商品搜索。

本节将介绍以上功能的具体内容和实现目的。

1．产品管理功能

产品管理实际上很简单，主要是通过操作数据库的记录，来完成插入、编辑、删除产品记录的功能。但是为了能使产品管理中的某些代码得到更好地复用，在规划产品管理的功能时，可以将在其他模块或在其他程序中可以复用的代码，单独以类的形式创建。

产品管理功能中涉及的，可以在其他模块或程序中使用的代码主要就是表格代码。通常情况下，要列出数据库中的相关记录，开发人员都会写一个表格，并根据要显示记录的实际情况确定表格的行和列，如列出文件和文件夹记录的操作。

因为表格的应用具有统一性，所以完全可以把表格共同的部分独立出来，以类的形式进行封装，从而达到代码复用的目的。

表格通用类在很多开发框架中都有实现，并且很多实现的方法都很优秀。本章介绍的表格实现方法，主要是向读者介绍一种思路，帮助读者在开发过程中解决问题。

产品管理涉及的图片管理，完成上传和显示的功能。

本章实现的购物网站中的产品管理，主要有产品名称、产品介绍、价格等内容，通过几个简单的字段演示产品管理的过程。读者也可以通过添加数据库中的字段和修改产品管理功能代码，添加更多的产品输入项。

2. 订单管理功能

订单管理的主要功能是把会员通过网站订购的产品，以列表的形式展示出来，并能够通过修改记录的状态，来区分未结算定单和结算定单之间的状态。

3. 会员管理

会员管理主要包括会员注册、登录等，以及会员登录后提供的结算功能。会员使用到的功能主要是购物车功能。

这里需要重点介绍购物车。购物车功能是购物网站一个重要的模块，购物车功能的提出，为会员减少了很多不必要的操作。在本章代码实现的章节，将创建一个可以复用的购物车类。

4. 商品搜索

产品搜索是购物网站一个重要的功能，从众多商品中快速地找到想要的商品，可以节省会员很多时间。本章实现的搜索功能，主要是根据用户输入的关键词，搜索与关键词相匹配的产品名称和介绍。读者可以为产品添加更多的属性（如型号、产地等字段），实现更多字段的查询。

21.1.3　文件目录结构设计及说明

在 b2c 文件夹下包含 17 个对象，其中，有 3 个文件夹和 14 个 PHP 应用脚本程序。下面具体介绍文件夹和程序的作用。

1. 平台根目录说明

在 b2c 文件夹（根目录）下包含 17 个对象，目录结构如图 21.3 所示。下面分别介绍每个文件的具体功能。

- index.php：在线购物平台首页（入口文件）。
- cart.php：购物车程序。
- global.php：全局配置文件。
- login.php：登录认证程序。
- product.php：产品列表程序。
- register.php：用户注册认证程序。
- search.php：产品搜索程序。
- store.php：产品详情显示程序。
- user.php：用户密码修改程序。

- ❑ user_order.php：用户订单管理程序。
- ❑ user_product.php：用户产品管理程序。
- ❑ user_product_upload.php：产品图片上传程序。
- ❑ class 文件夹：核心类程序文件夹。
- ❑ folder 文件夹：产品图片上传文件夹。
- ❑ templates 文件夹：页面模板文件存放文件夹。

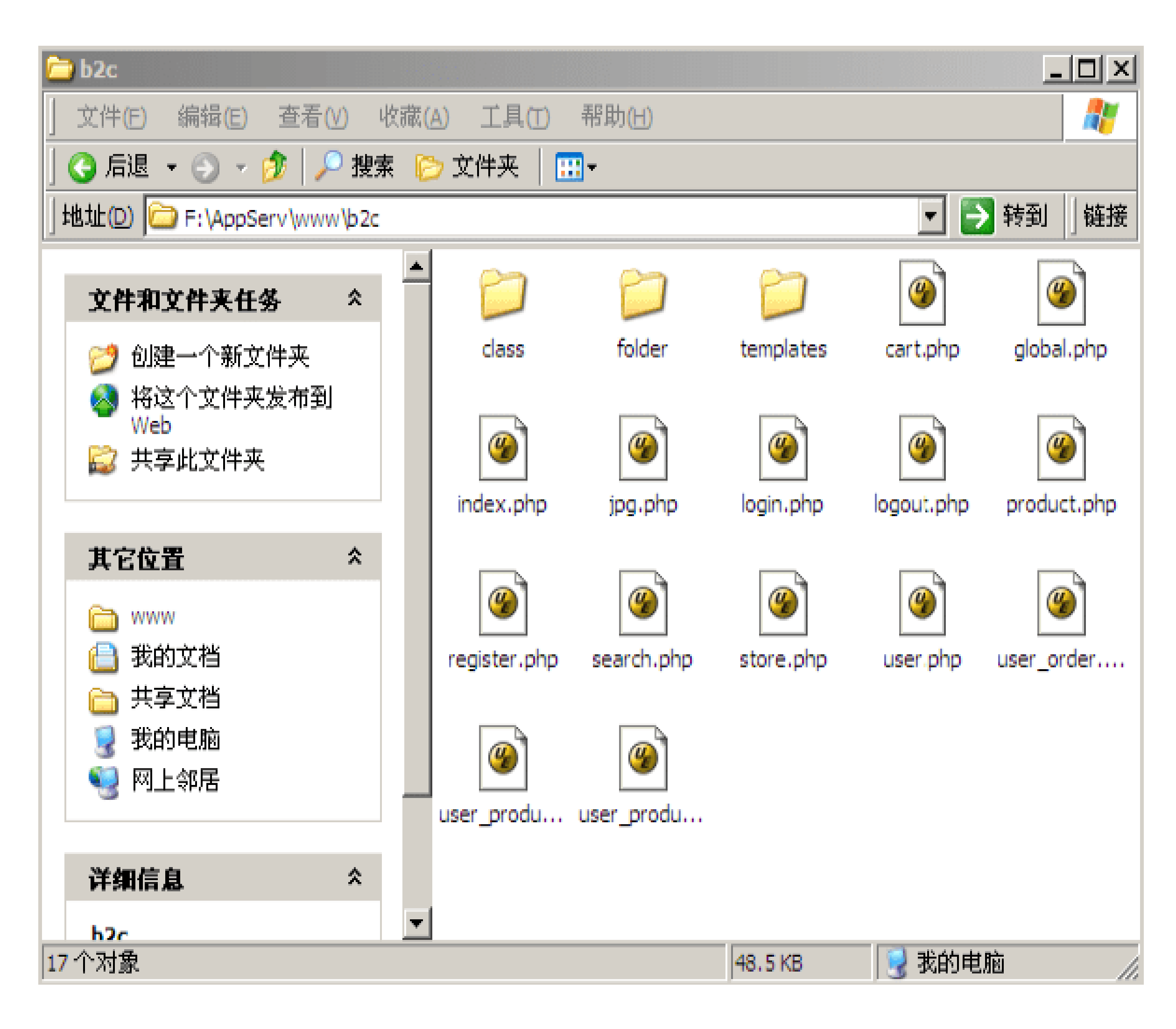


图 21.3　平台根目录

2．核心类程序文件夹

在 class 文件夹（核心类程序文件夹）下包含 5 个对象，目录结构如图 21.4 所示。下面分别介绍每个文件的具体功能。

- ❑ cart.class.php：购物车类程序。
- ❑ images.class.php：图像显示相关类程序。
- ❑ mysql.class.php：MySQL 数据库操作类程序。
- ❑ table.class.php：通用表格类程序。
- ❑ upload.class.php：文件上传类程序。

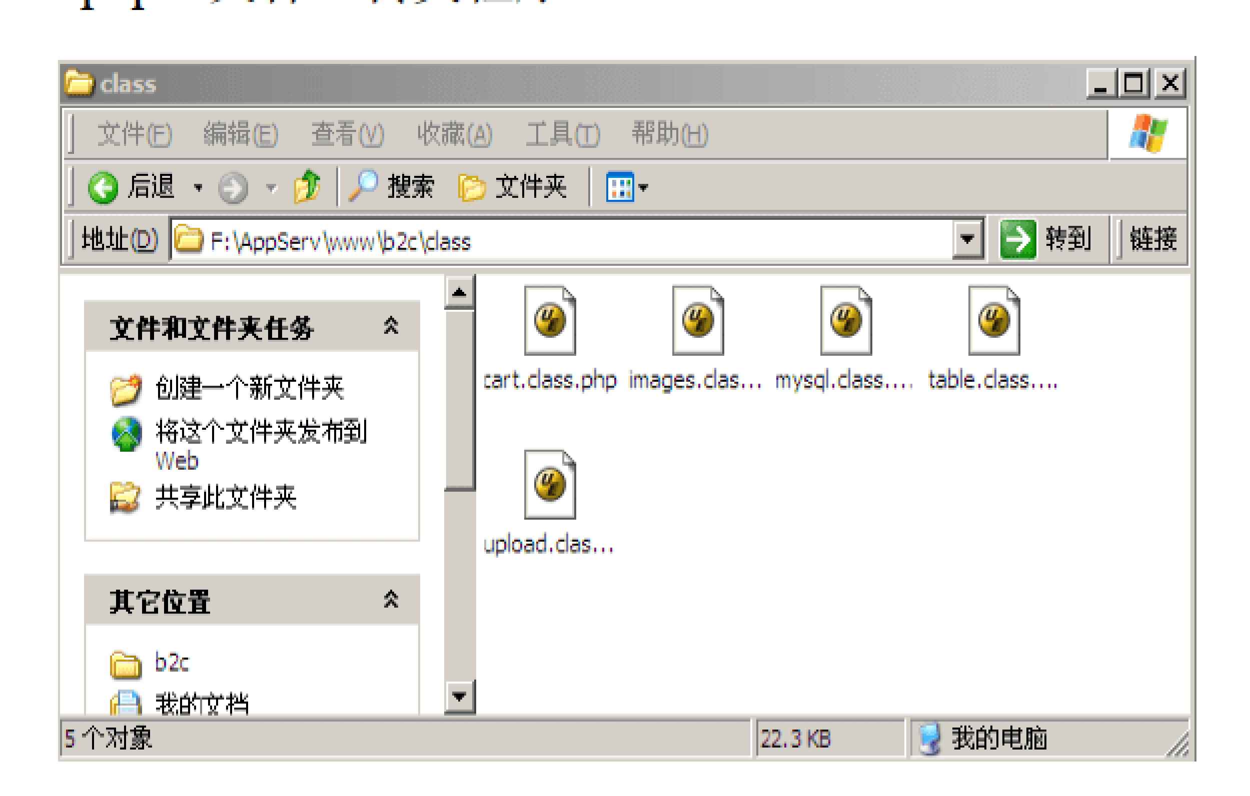


图 21.4　核心类程序文件夹

3．页面模板文件夹

在 templates 文件夹（页面模板文件夹）下包含 4 个对象，该文件夹用来存放页面模板相关文件，目录结构如图 21.5 所示。下面分别介绍每个文件的具体功能。

- Css 文件夹：存放页面对应的 css 样式文件。
- Flash 文件夹：存放页面模板对应的 flash 文件。
- Images 文件夹：存放页面模板对应的图片文件。
- Js 文件夹：存放页面对应的 JavaScript 文件。

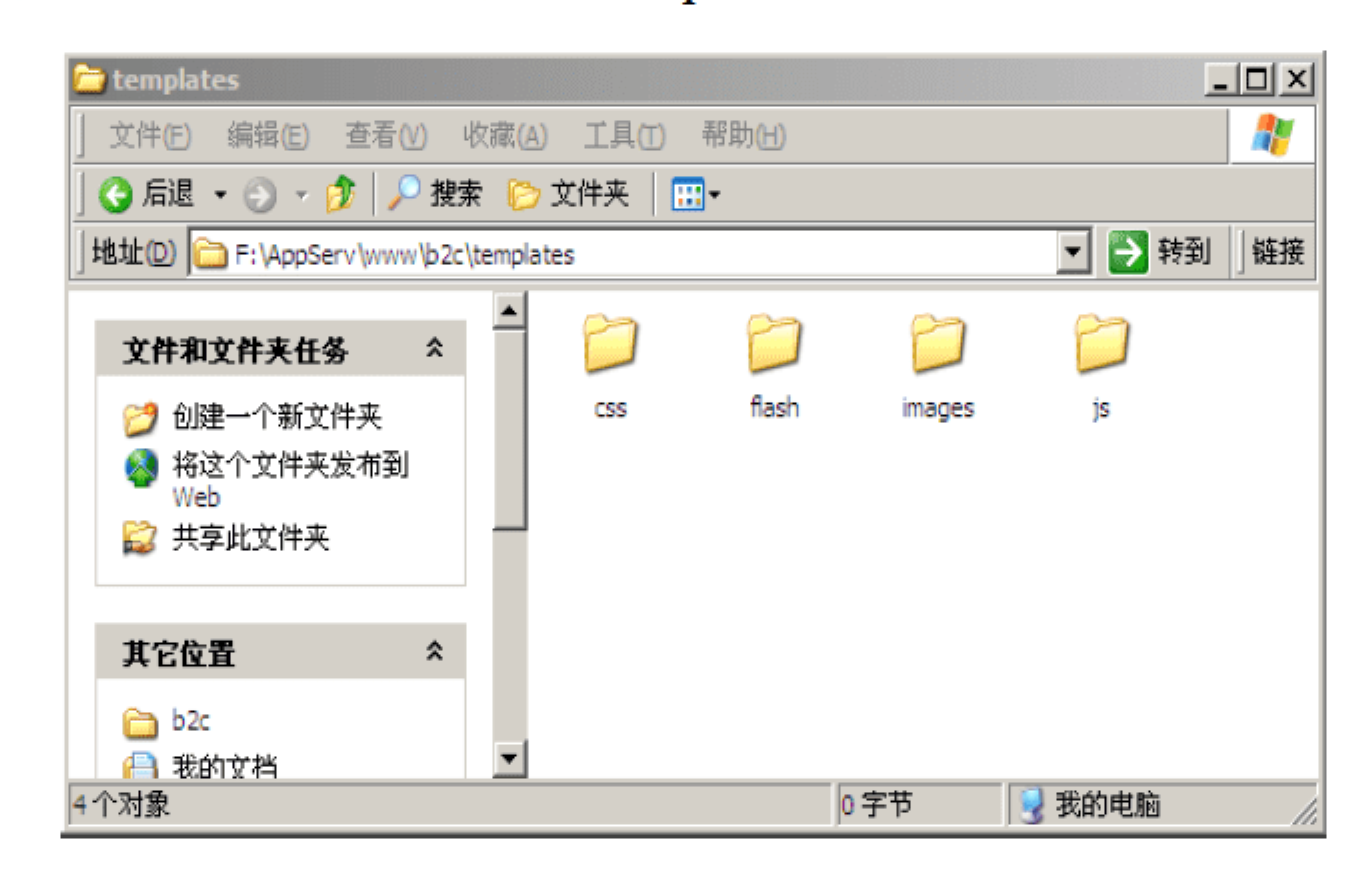


图 21.5　页面模板文件

21.2　数据库设计

在线购物平台是以为消费者提供在线购物为主要目的，因此数据库也围绕着这个目标设计。具体来说，需要完成如下目标：

（1）对产品的有效管理。

（2）用户对订单的有效管理。

（3）会员信息的管理。

（4）实现商品搜索匹配。

21.2.1　数据设计概述

根据网上购物最基本的要求，本章实现的购物网站需要创建 4 个数据表，分别是产品数据表、产品附件数据表、会员数据表、订单数据表。

下面具体介绍这 4 个数据表之间的关系。图 21.6 是在线购物平台数据库 E-R 关系图，直观地显示了这 4 个表之间的逻辑关系和各自的功能。

在线购物平台数据库表含义如下。

- 产品数据表（o_product）表：产品数据表用于记录管理员上传的产品信息，这些产品信息包括产品名称、产品介绍、市场价、商城价等信息。

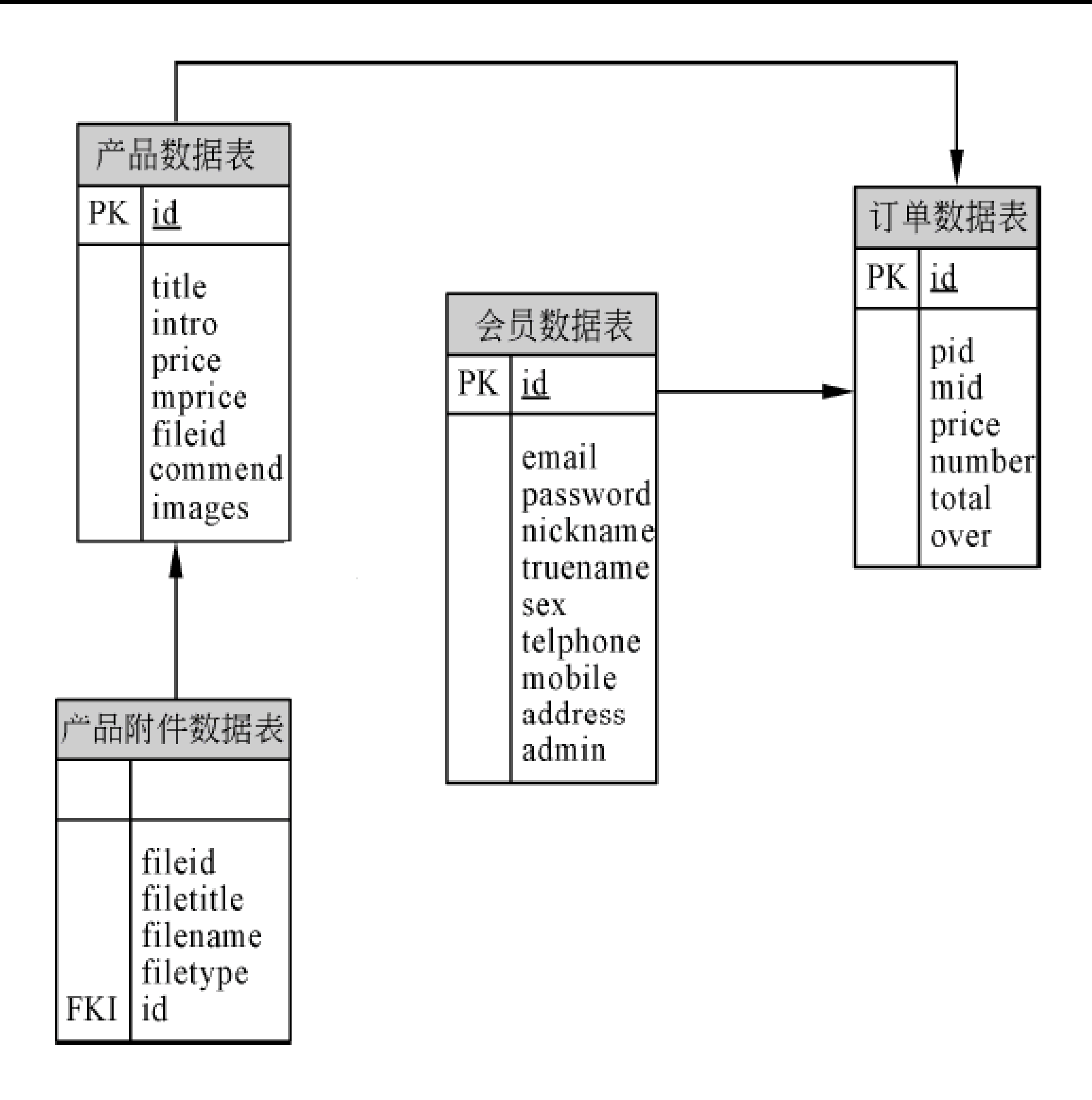


图 21.6　在线购物平台数据库 E-R 关系图

- ❑ 产品附件数据表（o_files）表：产品附件数据库，用于保存与产品相关的图片文件信息。
- ❑ 会员数据表（o_members）表：会员数据表，用于记录网站会员的登录，以及姓名、电话、地址等信息。
- ❑ 订单数据表（o_order）表：会员结算后，产生的购物清单保存在订单数据表中，便于管理人员查看会员已经购买的物品。

通过上面功能描述，可以清楚地了解到各个表的设计目的和需要完成功能。下面来看具体的设计步骤。创建名为 onlinestore 的数据库，SQL 语句如下：

```
CREATE DATABASE 'onlinestore';
```

21.2.2　产品数据表

在 onlinestore 数据库中，建立一个命名为 o_product 的产品数据表，各字段的详细含义如表 21.1 所示。创建产品数据表的 SQL 语句如下所示。

```
SET SQL MODE="NO AUTO VALUE ON ZERO";

--
-- 数据库: 'onlinestore'
--

-- --------------------------------------------------------

--
-- 表的结构 'o product'
--

CREATE TABLE 'o product' (
  'id' int(4) NOT NULL auto increment,
  'title' varchar(200) default NULL,
```

```
  'intro' text,
  'price' int(6) default NULL,
  'mprice' int(6) default NULL,
  'fileid' varchar(200) default NULL,
  'commend' varchar(10) default 'NO',
  'images' varchar(200) default NULL,
  PRIMARY KEY  ('id')
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

表 21.1　产品数据表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
Id	int	4	Not null	auto_increment	产品编号（自增主键）
Title	varchar	200	null		产品名称
Intro	text		null		产品介绍
Price	int	6	null		市场价格
Mprice	int	6	null		商城价格
Fileid	varchar	200	null		附件 id
commend	varchar	10	null		推荐
images	varchar	200	null		图片名称

注意：产品数据表核心的作用是记录产品的基本信息，这部分信息主要是用于产品的展示。由于在线购物类平台的特点，商品变动比较频繁，而且常常会增加额外的属性字段，因此这张表只记录产品的基础信息，其他的属性参数通过扩展表的方式关联。

在线购物平台程序初始化产品数据，执行以下 SQL 语句：

```
--
-- 导出表中的数据 'o product'
--

INSERT INTO 'o_product' VALUES (1, '加湿器', '健康环保加湿器', 158, 128, '',
'健康环保', NULL);
INSERT INTO 'o_product' VALUES (2, '足浴盆', '时尚、养生足浴盆', 258, 228, NULL,
'时尚、养生', NULL);
INSERT INTO 'o product' VALUES (3, '电吹风', '时尚小电器', 88, 68, NULL, '
新品推荐', NULL);
```

21.2.3　产品附件数据表

在 onlinestore 数据库中，建立一个命名为 o_files 的产品附件数据表，各字段的详细含义如表 21.2 所示。创建产品附件数据表的 SQL 语句如下所示。

```
--
-- 表的结构 'o_files'
--

CREATE TABLE 'o_files' (
  'id' int(4) NOT NULL auto increment,
  'fileid' varchar(200) default NULL,
```

```
  'filetitle' varchar(255) default NULL,
  'filename' varchar(200) default NULL,
  'filetype' varchar(100) default NULL,
  PRIMARY KEY  ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf-8 AUTO_INCREMENT=1 ;
```

表 21.2　产品附件数据表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
id	int	4	Not null	auto_increment	编号（自增主键）
fileid	varchar	200	null		产品 id
filetitle	varchar	255	null		附件标题
filename	Varchar	200	null		文件名称
filetype	varchar	200	null		文件类型

注意：产品附件数据库，用于保存与产品相关的图片文件信息，该表同时也可以用来扩展商品的其他参数属性。

在线购物平台程序初始化产品数据，执行以下 SQL 语句：

```
--
-- 导出表中的数据 'o files'
--

INSERT INTO 'o files' VALUES (1, '3bace75f7077eebea49b2718d31a676c', '数码相机', '数码相机', '图片');
```

21.2.4　会员数据表

在 onlinestore 数据库中，建立一个命名为 o_members 的会员数据表，各字段的详细含义如表 21.3 所示。创建会员数据表的 SQL 语句如下所示。

```
--
-- 表的结构 'o_members'
--

CREATE TABLE 'o members' (
  'id' int(4) NOT NULL auto increment,
  'email' varchar(200) default NULL,
  'password' varchar(200) default NULL,
  'nickname' varchar(100) default NULL,
  'truename' varchar(100) default NULL,
  'sex' int(4) default NULL,
  'telphone' varchar(100) default NULL,
  'mobile' varchar(100) default NULL,
  'address' text,
  'admin' int(4) default '0',
  PRIMARY KEY  ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

表 21.3　会员数据表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
id	int	4	Not null	auto_increment	会员编号（自增主键）
Email	varchar	200	null		注册 Email
Password	varchar	255	null		会员密码
Nickname	Varchar	100	null		昵称
truename	varchar	100	null		真实姓名
sex	int	4	null		性别
telphone	varchar	100	null		固定电话
Mobile	varchar	100	null		移动电话
Address	text		null		通讯地址
admin	int	4	null		是否为管理员

注意：会员数据表，用于记录网站会员的登录，以及姓名、电话、地址等信息。读者可以根据项目的实际情况设定必需项。例如，购物类站点真实姓名、会员密码、移动电话和通讯地址就是必选项（因为涉及下线收货）。

21.2.5　订单数据表

在 onlinestore 数据库中，建立一个命名为 o_order 的订单数据表，各字段的详细含义如表 21.4 所示。创建订单数据表的 SQL 语句如下。

```
--
-- 表的结构 'o_order'
--

CREATE TABLE 'o order' (
  'id' int(4) NOT NULL auto increment,
  'pid' int(4) default NULL,
  'mid' int(4) default NULL,
  'price' int(6) default NULL,
  'number' int(4) default NULL,
  'total' int(6) default NULL,
  'over' int(4) default '0',
  PRIMARY KEY  ('id')
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

表 21.4　订单数据表

字　段	数 据 类 型	长　度	NULL	默 认 值	字 段 说 明
id	int	4	Not null	auto_increment	订单编号（自增主键）
Pid	int	4	null		商品 id
Mid	int	4	null		会员 id
Price	int	6	null		价格
Number	int r	4	null		数量
Total	int	6	null		总价
over	int	4	null		是否购买

注意：会员结算后，产生的购物清单保存在订单数据表中，便于管理人员查看会员已经购买的物品。

21.3　核心程序说明

在线购物平台程序的核心是几个功能类文件，主要是涉及独立的业务逻辑和处理数据库的数据交互的底层程序，这部分使用类文件的方式实现。在后面的具体在线购物平台的业务逻辑实现部分，就可以"专注"业务逻辑本身的实现，而且不必考虑底层的代码实现。在本章程序中主要涉及如下几个核心程序模块：

- 通用表格类：格式化待处理的数据以表格的方式输出。
- 购物车类：处理购物过程中的产品存储逻辑。
- 文件上传类：主要处理图像上传（缩略图）。
- 图片显示相关类：处理与图片显示有关的逻辑。
- 公用和配置文件：公用文件和配置文件。

21.3.1　通用表格类

当程序在处理大量数据时，往往会用到表格，在本章要实现的购物网站中，使用表格的地方很多，例如产品列表、购物清单列表、订单列表等。程序根据各种列表的要求，创建一个通用的表格类，来完成购物网站中关于表格输出的要求。

要创建一个可以用于多种情况下的类，要先考虑类需要完成的功能。本小节需要实现的表格类，需要完成以下功能。

- 记录列表显示：根据参数，以表格的形式，列出记录内容。
- 列表头：根据参数，创建一个表头，以帮助用户理解数据。
- 表单：使用一个表单，来管理表格中的表单控制。
- 与列表相关的表单控件：将表格记录与表单控件进行绑定，在提交表单时，用来获取记录的相关数据。
- 与表单提交相关的表单控制：当表单提交时，根据这个控制来判断需要调用的代码。
- 工具栏：根据参数创建一行工具栏，生成的工具栏按钮，可以控制表单提交的动作等内容。
- JavaScript 支持：为了能够更加灵活地控制表单控件，加入 jQuery 框架支持，完成其中与表单相关的 JavaScript 代码操作。

注意：jQuery 是一个快速、简洁的 JavaScript 库，使用户能更方便地处理 HTML documents、events、实现动画效果，并且方便地为网站提供 AJAX 交互。

在规划好需要的功能后，就可以开始创建通用表格类的代码了。通用表格类的代码保存在"./class"目录下，文件名为 table.class.php。详细代码如下所示。

```
<?php
class table {
    public $_jspath = "templates/js/";          //保存 JavaScript 文件的目录
    public $_name = "";                         //保存名称
    public $_action = "";                       //保存动作变量
    function table($name = "tableForm") {
        $this->_name = $name;
    }
    function normal($header, $data = NULL, $control = false, $toolbar = NULL,
    $width = '100%') {
        $html = $this->js ( $toolbar );         //加载 JavaScript 文件
        $html .= '<form name="' . $this->_name . '" id="' . $this->_name .
        '"method="post" action="' . $this->_action . '">';
        if ($toolbar != NULL) {
            foreach ( $toolbar as $k => $v ) { //遍历工具栏数组，创建工具栏
                $html .= '<input type="button" name="' . $k . '" id="' . $k .
                '" value="' . $v ["value"] . '">';
            }
        }
        $html .= '<table class="tableList" style="width:' . $width . '">';
        $html .= '<tr>';
        if ($control == true) {                 //根据参数，决定是否显示表单控制
            $html .= '<th><input type="checkbox" name="toggle" id="toggle"
            value="" onClick="' . $this->_name . '_checkAll(' . (count
            ( $data )). ');" /></th>';
        }
        foreach ( $header as $v ) {
            $html .= "<th>" . $v . "</th>";     //输出表头内容
        }
        $html .= "</tr>";
        $rs = 0;
        $cb = 0;
        if ($data != NULL) {
            if (is_object ( $data )) {
                $data = ( array ) $data;        //转换数据为数组
            }
            foreach ( $data as $k => $v ) {
                //兼容性处理开始
                if (is_object ( $v )) {
                    $v = ( array ) $v;  //如果传入的数据是对象，将其转换为数组
                }
                //如果要处理的数据不是数组，跳过此次循环
                if (gettype ( $v ) != "array") {
                    continue;
                }
                //使用 array_values 将关联数组转换为数字为键名的数组
                $v = array_values ( $v );
                //兼容性处理结束
                if ($rs == 2) {
                    $rs = 0;
                }
                $html .= '<tr class="row' . $rs . '" id="' . $k . '">';
                if ($control == true) { //允许显示表格控制时，输出表格控制内容
                    $html .= '<td><input type="checkbox" id="cb' . $cb . '"
                    name="cid[]" value="' . $v [0] . '" onclick="' .
                    $this->_name . '_isChecked(this.checked);" /></td>';
                }
                for($i = 0; $i < count ( $v ); $i ++) {
                    $html .= "<td>" . $v [$i] . "</td>";
```

```
                }
                $html .= "</tr>";
                $rs ++;
                $cb ++;
            }
        }
        $html .= '</table>';
        $html .= '<input type="hidden" name="boxchecked" id="' . $this->
        name . ' boxchecked" value="0"><input type="hidden" name="do" id="' .
        $this-> name . ' do" value=""></form>';
        return $html;
    }
    //JS 判断函数
    function js($toolbar = NULL) {
        //加载 jQuery 文件
        $js = "<script type='text/JavaScript' src='" . $this-> jspath .
        "jquery-1.2.6.js'></script>";
        $js .= '<script type="text/JavaScript">';
        if ($toolbar != "") {
            $js .= '$(document).ready(function(){';
            foreach ( $toolbar as $k => $v ) {      //循环表格标签变量
                $js .= '$("#' . $k . '").click(function(){
                    $("#' . $this->_name . '").attr("action","' . $v
                    ["action"] . '");
                    $("#' . $this-> name . ' do").val("' . $k . '");
                    if($("#' . $this-> name . ' boxchecked").val()==0){
                        alert("请选择要处理的记录!");
                    }else{
                        $("#' . $this-> name . '").submit();    //发送请求
                    }
                });';
            }
            $js .= "});";
        }
        //创建与工具栏有关的 JavaScript 代码
        $js .= "
        function " . $this->_name . "_checkAll( n, fldName ) {
            if (!fldName) {
               fldName = 'cb';
            }
            var f = document." . $this-> name . ";
            var c = f.toggle.checked;
            var n2 = 0;
            for (i=0; i < n; i++) {
                cb = eval( 'f.' + fldName + '' + i );
                if (cb) {
                    cb.checked = c;
                    n2++;
                }
            }
            if (c) {
                document." . $this->_name . ".boxchecked.value = n2;
            } else {
                document." . $this-> name . ".boxchecked.value = 0;
            }
        }
        function " . $this-> name . " isChecked(isitchecked){
            if (isitchecked == true){
                document." . $this-> name . ".boxchecked.value++;
            }
```

```
            else {
                document." . $this-> name . ".boxchecked.value--;
            }
            if(document." . $this->_name . ".boxchecked.value == 0){
                document." . $this-> name . ".toggle.checked = false;
            }
        }
        </script>";
        return $js;
    }
}
?>
```

【代码解读】

（1）表单类创建后，首先定义了 3 个公共变量，这 3 个变量分别是：

- $_jspath：用于存放 jQuery 框架的路径，用户可以读取和修改这个变量的内容。
- $_name：用于定义表单的 name 属性，防止在同一页内出现相同的表格，类实例化时这个变量被修改，默认值为“tableForm”。用户可以读取和修改这个变量的内容。
- $_action：用于设置表单默认的 action 属性，默认值为空。用户可以读取和修改这个变量的内容。

（2）table()函数：构造函数，用于在类实例化时，修改公共变量$_name 的值。该函数只有一个参数，默认值为“tableForm”。

（3）normal()函数：创建表单与表格内容的主体函数，它有如下 5 个参数。

- 第一个参数是必选参数，用于指定创建表头的数据，其参数类型是数组。
- 第二个参数是可选参数，用于指定创建表格内容的数据，其参数类型是数组。为了兼容参数类型，代码中进行了兼容性设置。当代码中的参数是对象时，将这些对象转换成数组；当代码中的变量类型不是数组时，将跳过这个变量，不进行处理。
- 第 3 个参数是可选参数，当其值为 false 时，不显示表格中的表单控件。当其值为 true 时，将显示表格中的表单控件，并与表格数据进行绑定。
- 第 4 个参数是可选参数，当其值为 NULL 时，不显示工具栏控制；当其值不为 NULL 时，将根据参数的内容，创建工具栏按钮。
- 第 5 个参数是可选参数，用于控制表格的宽度，默认值为 100%。

（4）normal()函数在开始时，调用了同类中的唯一一个函数 js()函数。js()函数创建一段引用 jQuery 框架代码的语句，并将根据参数，创建一段与工具栏有关的 JavaScript 代码，以完成工具栏按钮控制表单的操作。

（5）js()函数中除了完成与工具栏相关的代码外，还实现了两个 JavaScript()函数，用于实现表格控制的全选功能。

（6）为了防止同一页中出现相同的 JavaScript()函数，js()函数中，使用了$_name 变量创建相同内容，但不同函数名称的 JavaScript 代码。

本节规划的功能，只是满足购物网站的需要。读者如果感兴趣，可以使表格实现更多的显示功能，例如在表格中显示图片、表单控件等内容。

注意：在后面的章节中，都使用通用表格类来表述 table.class.php。

21.3.2　购物车类

购物车主要是用于临时记录会员购买物品的名称、数据和价格。本小节将使用 SESSION 记录购物清单，来完成购物车功能。

购物车主要包括以下功能：

- 添加记录：向购物数据中添加一条记录。如果该条记录存在，将更新记录的数量。
- 删除记录：根据参数删除一条购物记录。
- 统计：计算购物车中物品的总价格。
- 返回购物内容数组：返回的数组将用于其他控制，例如通用表单类。
- 读取数据：从 SESSION 中读取购物记录。
- 保存数据：将购物记录保存到 SESSION 中。
- 清除数据：清除购物车中的所有数据。

在规划好需要的功能后，就可以创建购物车类的代码了。购物车类的代码保存在./class 目录下，文件名为 cart.class.php，详细代码如下所示。

```
<?php
//创建购物车类
class cart {
    //向数组中添加各记录产品信息的数据
    function addItem($id, $title, $number, $price) {
        $temp = ( array ) $this->restore ();   //读取 SESSION 中的产品数据
        if (array key exists ( $id, $temp )) {
                                        //当产品已经存在时，只修改产品数量
            $temp [$id] ["number"] = $temp [$id] ["number"] + $number;
        } else {
            //产品不存在时，新建一条记录
            $temp [$id] ["id"] = $id;
            $temp [$id] ["title"] = $title;
            $temp [$id] ["number"] = $number;
            $temp [$id] ["price"] = $price;
        }
        //将数据保存到 SESSION 中
        $this->store ( $temp );
    }
    //根据 ID，从数组中删除一条记录
    function removeItem($id) {
        $temp = $this->restore ();          //将产品记录读取到变量$temp 中
        if (is array ( $temp [$id] )) {     //判断变量是否为数组
            unset ( $temp [$id] );          //删除指定的数组内容
        }
        $this->store ( $temp );             //保存删除元素后的数组内容
    }
    //统计总的价格
    function  total() {
        $total = "";
        $temp = $this->restore ();          //将购物车记录保存到$temp 变量中
        if (is array ( $temp ) and count ( $temp ) > 0) {
            foreach ( $temp as $v ) {
                $total += $v ["number"] * $v ["price"];//获得产品总的价格
            }
```

```
        }
        return $total;
    }
    //以数组形式返回购物车的内容
    function listArray() {
        $temp = $this->restore ();
        return $temp;                           //以数组形式返回购物车记录
    }
    //恢复数据
    function restore() {
        //从 SESSION 数据中，读取购物车数据
        $cart_items = $_SESSION ["cart_items"];//获取 SESSION 中的购物车记录
        $items = unserialize ( stripslashes ( $cart_items ) );
                                                //将 SESSION 中的数据反序列化
        //返回数据
        return $items;
    }
    //保存数据
    function store($items) {
        $items = serialize ( $items );          //将购物车数据序列化
        $_SESSION ["cart_items"] = $items;      //保存到 SESSION 中
    }
    //清空数据
    function clear() {
        unset ( $_SESSION ["cart_items"] );     //删除 SESSION 中购物车的记录
    }
}
?>
```

【代码解读】

（1）addItem()函数：使用 restore()函数，从 SESSION 中恢复购物记录。向购物数据中添加一条记录，当购物记录中存在相同的记录时，只更新这条记录的数量。完成后使用 store()函数保存最新的购物记录。

（2）removeItem()函数：根据提供的参数，使用 unset()函数，删除购物记录中的一条。

（3）_total()函数：用于统计购物车中所有物品的总价。

（4）listArray()函数：将 SESSION 中记录的购物数据，以数组的形式返回，供其他模块使用。

（5）store()函数：将购物记录序列化后，保存到 SESSION 中。

（6）restore()函数：读取 SESSION 中关于购物记录的数据，并反序列化数据后返回购物记录。

本节实现的购物车功能比较简单，为了提升客户体验，读者也可以尝试使用之前介绍过的 AJAX 模式，完成购物车相关的功能。

21.3.3　文件上传类

文件上传类（upload.class.php）的核心功能是处理文件上传相关的业务逻辑，该类文件由以下几个部分构成。

- 文件上传方法 uploadFile()（多个文件上传使用 multiUpload()方法）。
- 获取文件名后缀方法 getExtendedName()。

- 文件类型验证方法 allowType()。
- 创建上传文件目录方法 createDir()。
- 创建 JPG 格式的缩略图方法 thumbnail()。

定义文件上传类和两个私有方法，代码如下：

```
class upLoad {
    private $ thumb width = 100;
    private $ thumb height = 100;
}
```

1. 文件上传方法

文件上传部分，首先介绍处理单个文件上传逻辑的方法 uploadFile()，该方法是从表单中获得文件的基本信息，然后将上传的临时文件保存到服务器指定的路径中，并返回状态。代码如下：

```
function uploadFile($fileField,$userid,$oldPath="") {
    $RESIZEWIDTH = $this->_thumb_width;         //缩略图宽度
    $RESIZEHEIGHT = $this->_thumb_height;       //缩略图高度
    $upfile = "";
    $files = $_FILES [$fileField];              //获取$_FILES 变量中的数据
    $fileName = $files ['name'];                //获取文件名
    $fileType = $files ['type'];                //获取文件类型
    //获取临时文件的文件名
    $fileTemp = $files ['tmp name'];
    if($fileName != "" and $fileTemp != "" and $fileType != ""){
        if($this->allowType($fileType)){
            //获取文件大小
            $upfile["filesize"] = filesize($fileTemp);
            //创建文件夹，将上传的文件保存到新创建的文件夹中
            if($oldPath==""){
                $filePath = $this->createDir($userid);
            }else{
                $filePath = "./folder/";        //设置文件路径
            }
            //获取文件扩展名
            $fileExtendedName = $this->getExtendedName ( $fileName );
            //设置新的文件名称，以保证上传的文件不重名
            $newFileName = time (). "." . $fileExtendedName;
            //使用 move uploaded file()函数，上传的临时文件，保存到服务器指定的路
            径中，并返回状态
            $upfile["filename"]= $newFileName;
            $upfile["filetype"]= $fileType;
            $upfile["filestat"] = @move_uploaded_file ( $fileTemp,
            $filePath . $newFileName ) ? "true":"false";
            switch($fileType){          //判断文件类型
                case "image/pjpeg":
                    $im = imagecreatefromjpeg ( $filePath.$newFileName );
                break;
                case "image/x-png":
                    $im = imagecreatefrompng ( $filePath.$newFileName );
                break;
                case "image/gif":
                    $im = imagecreatefromgif ( $filePath.$newFileName );
                break;
```

```
            }
            //创建 JPG 格式的缩略图
            $this->thumbnail( $im, $RESIZEWIDTH, $RESIZEHEIGHT,  $filePath.
            "".$newFileName );
            ImageDestroy ( $im );  //销毁图像，释放与其关联的内存
        }else{
            $upfile["filename"]= "非法的文件类型。";
            $upfile["filestat"] = "false";
        }
    }else{
        $upfile["filename"]= "无效的文件数据。";
        $upfile["filestat"] = "false";
    }
    return $upfile;
}
```

【代码解读】

文件上传方法的程序实现流程如下：

（1）设置基础参数（缩略图尺寸）。

（2）获得文件本身参数信息（文件名、文件类型等）。

（3）设置文件存储路径、文件名称排重。

（4）使用 move_uploaded_file()函数，上传的临时文件，保存到服务器指定的路径中，并返回状态。

（5）判断上传文件类型。

（6）创建 JPG 格式的缩略图，完成上传。

多文件上传的实现逻辑是单文件上传模式的扩展，本质上是将文件通过循环遍历的方式，逐个上传到指定的目录中。该方法的代码如下：

```
function multiUpload($fileField,$userid,$oldPath="") {
    $RESIZEWIDTH = $this->_thumb_width;            //缩略图宽度
    $RESIZEHEIGHT = $this-> thumb height;          //缩略图高度
    $upfiles = "";
    //获取$ FILES 变量中的数据
    $files = $_FILES [$fileField];
    $fileName = $files ['name']; //获取上传文件的文件名
    $fileTemp = $files ['tmp_name'];               //获取临时文件的文件名
    $fileType = $files ['type'];                   //获取文件类型
    //创建用于保存文件的文件夹
    if($oldPath==""){
        $filePath = $this->createDir($userid);
    }else{
        $filePath = $oldPath;
    }
    //遍历上传的文件内容
    for($i = 0; $i < count ( $fileName ); $i ++) {
        //如果文件名与临时文件都不为空，则上传文件
        if ($fileName [$i] != "" and $fileTemp [$i] != "" and $fileType
        [$i]!="" ) {
            if($this->allowType($fileType[$i])){
                //设置新文件名称
                $newFileName = time () . $i;
                $fileExtendedName = $this->getExtendedName ( $fileName
                [$i] );                               //设置文件类别
```

```
                $newFilePathAndName = $newFileName . "." . $fileExtended-
                Name;
                $upfiles["filesize"][$i] = filesize($fileTemp[$i]);
                $upfiles["filename"][$i] = $newFilePathAndName;
                $upfiles["filetype"][$i] = $fileType[$i];
                $upfiles["filestat"][$i] = @move uploaded file ( $fileTemp
                [$i], $filePath.$newFilePathAndName ) ? "true" : "false";
                switch($fileType[$i]){          //判断上传文件类型
                    case "image/pjpeg":
                        $im = imagecreatefromjpeg ( $filePath.$newFile-
                        PathAndName );
                    break;
                    case "image/x-png":
                        $im = imagecreatefrompng ( $filePath.$newFilePath-
                        AndName );
                    break;
                    case "image/gif":
                        $im = imagecreatefromgif ( $filePath.$newFilePath-
                        AndName );
                    break;
                }
                //创建 JPG 格式的缩略图
                $this->thumbnail( $im, $RESIZEWIDTH, $RESIZEHEIGHT,
                $filePath. "/thumbnail/".$newFilePathAndName);
                ImageDestroy ( $im );       //销毁图像释放与 image 关联的内存
            }else{
                $upfiles["filename"][$i] = "非法的文件类型。";
                $upfiles["filestat"][$i] = "false";
            }
        }else{
            $upfiles["filename"][$i] = "无效的文件数据。";
            $upfiles["filestat"][$i] = "false";
        }
    }
    return $upfiles;
}
```

【代码解读】

多文件上传的代码实现是以单文件上传为实现基础的。文件上传的实现方法是相同的，不同的方面是对待上传文件异常情况的判断处理和循环遍历待上传文件数组。

注意： ImageDestroy ();函数的作用是销毁图像释放与 image 关联的内存。在文件的最后使用该函数销毁 image 关联的内存，可以防止数据污染和异常状态下造成的重复上传操作，减少可能的文件冗余。

2．获取文件名后缀方法

下面介绍获取文件名后缀方法 getExtendedName()，该方法是将文件名字符串切割，“.”后面的就是文件的后缀名。该方法代码如下：

```
/**
 * 取得文件名后缀
 */
function getExtendedName($fileName) {
    return end ( explode ( '.', $fileName ) );
}
```

3．文件类型验证方法

文件类型验证方法 allowType()，该方法的作用是检查是否为允许上传的文件类型。具体代码如下：

```
/**
 * 检查是否为允许上传的文件类型
 */
function allowType($type) {
    //设置不允许上传文件类型数组
    $types = array ('application/x-js', 'application/octet-stream',
    'application/x-php','text/html' );
    if (in array ( $type, $types )) {
        return FALSE;
    } else {
        return TRUE;
    }
}
```

【代码解读】

文件类型验证方法 allowType()通过设置不允许上传文件类型数组，然后通过 in_array()函数判断当前文件是否在数组中，从而实现判断的目的。

这里预置了 4 种最常用到的数据类型，读者可以在此处添加额外的文件类型数组，实现文件类型检查功能。

4．创建上传文件目录方法

创建上传文件目录方法 createDir()，该方法的作用是将当前用户上传的文件生成指定（规定格式）的目录，并设置目录权限。该方法的具体代码如下：

```
/**
 * 建立目录 目录格式：用户 ID/年/月/日/
 */
function createDir($userid) {
    $root = 'folder';                           //定义上传文件根目录
    $pathSign = DIRECTORY SEPARATOR;
    $u = $userid . $pathSign;
    $y = date ( 'Y' ) . $pathSign;              //获得日期参数
    $m = date ( 'm' ) . $pathSign;
    $d = date ( 'd' ) ;
    $realpath = $root . $pathSign . $u . $y . $m . $d;
    $path = $root . $pathSign . $u . $y . $m . $d .$pathSign."thumbnail";
    $dirArray = explode ( $pathSign, $path );  //获得目录数组
    $tempDir = '';
    foreach ( $dirArray as $dir ) {             //循环遍历目录数组
        $tempDir .= $dir . $pathSign;
        $isFile = file exists ( $tempDir );
        clearstatcache ();
        if (! $isFile && ! is dir ( $tempDir )) {
            @mkdir ( $tempDir, 0777 ); //设置当前目录权限
        }
    }
    return $realpath . $pathSign;
}
```

【代码解读】

创建上传文件目录方法的程序实现流程如下：

（1）定义上传文件的根目录（起点）。

（2）获得当前日期参数（拼接目录文件名）。

（3）获得目录数组。

（4）循环生成目录并设置对应的文件夹权限。

5．创建JPG格式的缩略图方法

创建 JPG 格式的缩略图方法 thumbnail():

```
function thumbnail($im, $maxwidth, $maxheight, $name) {
        $width = imagesx ( $im );      //设置图片宽度
        $height = imagesy ( $im );     //设置图片高度
        if (($maxwidth && $width > $maxwidth) || ($maxheight && $height >
        $maxheight)) {
            if ($maxwidth && $width > $maxwidth) { //判断图片尺寸是否符合
                $widthratio = $maxwidth / $width;
                $RESIZEWIDTH = true;
            }
            if ($maxheight && $height > $maxheight) {  //最大高度验证
                $heightratio = $maxheight / $height;
                $RESIZEHEIGHT = true;
            }
            //实际图片尺寸验证
            if ($RESIZEWIDTH && $RESIZEHEIGHT) {
                if ($widthratio < $heightratio) {
                    $ratio = $widthratio;
                } else {
                    $ratio = $heightratio;
                }
            } elseif ($RESIZEWIDTH) {
                $ratio = $widthratio;
            } elseif ($RESIZEHEIGHT) {
                $ratio = $heightratio;
            }
            //设置生成图片尺寸
            $newwidth = $width * $ratio;
            $newheight = $height * $ratio;
            if (function exists ( "imagecopyresampled" )) {
                //创建一个真彩色图像
                $newim = imagecreatetruecolor ( $newwidth, $newheight );
                imagecopyresampled ( $newim, $im, 0, 0, 0, 0, $newwidth,
                $newheight, $width, $height );
            } else {
                $newim = imagecreate ( $newwidth, $newheight );
                imagecopyresized ( $newim, $im, 0, 0, 0, 0, $newwidth,
                $newheight, $width, $height );
            }
            ImageJpeg ( $newim, $name);// JPEG 格式将图像输出到浏览器或文件
            ImageDestroy ( $newim );
        } else {
            ImageJpeg ( $im, $name . ".jpg" );
                                            // JPEG 格式将图像输出到浏览器或文件
        }
    }
```

【代码解读】

创建 JPG 格式的缩略图方法 thumbnail()的实现原理，是通过 imagecreatetruecolor()函数创建一个指定尺寸的图片单元，然后通过 ImageJpeg()将生成的 JPEG 图像输出到浏览器或生成文件，完成创建缩略图文件。

21.3.4　图片显示相关类

图片显示类的功能是负责处理图像显示相关的逻辑，该类文件实现的功能有：

- 幻灯片模式显示图片（拥有开始和暂停按钮，单击放大）。
- 图片的滑动显示。
- 图片网格列表显示。
- 单个图片单击放大显示（带有遮罩效果）。

图片显示类的具体代码如下：

```
<?php
class images{
    /**
    * 幻灯片模式显示图片
    * 拥有开始和暂停按钮，单击放大
    * **/
    function lantern($folder,$images){
?>
<link rel="stylesheet" type="text/css" media="screen" href="templates/css/
photoslider.css" />
<script type="text/javascript" src="templates/js/jquery-1. 2.6.js"></script>
<script type="text/javascript" src="templates/js/photoslider.js"></script>
<div class="photoslider" id="default"></div>

<script type="text/javascript">
$(document).ready(function(){
    //初始化文件目录与缩略图目录
    FOTO.Slider.baseURL = ".";
    FOTO.Slider.fileURL = '<?php echo $folder;?>';
    FOTO.Slider.thumbFileURL = '<?php echo $folder;?>/thumbnail';
    FOTO.Slider.bucket = {
        'default': {
<?php
$js array = "";
$li array = "";
foreach($images as $k=>$v){
    $js array[] = '"'.$v[2].'"';
    $li array[]= '"'.$v[2].'"'.": {'thumb': '".$folder."/thumbnail/".$v[2]
    .".jpg.jpg', 'main': '".$folder."/".$v[2]."', 'caption': '".$k.$v
    [4]."'}";
}
echo implode(",",$li array);
?>
        }
    };
    var ids = new Array(<?php echo implode(",",$js array);?>);
    FOTO.Slider.importBucketFromIds('default',ids);
    FOTO.Slider.reload('default');
    FOTO.Slider.preloadImages('default');
    FOTO.Slider.enableSlideshow('default');
```

```
});
</script>
<?php
    }
    //滑动显示
    function slideview($folder,$images,$width=600,$height=500){
?>
<link rel="stylesheet" type="text/css" media="screen" href="templates/css/
slideview.css" />
<script type="text/javascript" src="templates/js/jquery-1.2.6.js"></script>
<script type="text/javascript" src="templates/js/jquery.easing.1.3.js">
</script>
<script type="text/javascript" src="templates/js/jquery.slideviewer.1.1.
js"></script>
<div id="mygalone" class="svw">
     <ul>
<?php
foreach($images as $k=>$v){
    echo '<li><img alt="'.$k.".".$v[4].'"  src="'.$folder."/".$v[2].'"
    width="'.$width.'" height="'.$height.'" /></li>';
}
?>
     </ul>
</div>
<script type="text/javascript">
    $(window).bind("load", function(){
        $("div#mygalone").slideView()
    });
</script>
<?php
    }
    //网格列表显示
    function gridview($folder,$images){
?>
<link rel="stylesheet" type="text/css" media="screen" href="templates/
css/gridview.css" />
<script type="text/javascript" src="templates/js/jquery-1.2.6.js">
</script>
<script type="text/javascript" src="templates/js/jqGalViewIII.js">
</script>
<script type="text/javascript">
    $(document).ready(function(){
        $('#Gallery').jqGalViewIII();
    });
</script>
    <ul title="My Gallery" id="Gallery">
<?php
foreach($images as $k=>$v){
    echo '<li><a href="'.$folder."/".$v[2].'"><img src="'.$folder."/
    thumbnail/".$v[2].'.jpg" alt="'.$k.".".$v[4].'" width="72" height=
    "48" border="0"/></a></li>';
}
?>
    </ul>
<?php
    }
    //单个图片单击放大显示，带有遮罩效果
    function singleShow($folder,$images,$width=100,$height=100){
?>
<link rel="stylesheet" type="text/css" media="screen" href="templates/css/
singleshow.css" />
```

```
<script type="text/javascript" src="templates/js/jquery-1.2.6.js">
</script>
<script src="templates/js/iutil.js" type="text/javascript"></script>
<script src="templates/js/ifx.js" type="text/javascript"></script>
<script src="templates/js/ifxslide.js" type="text/javascript"></script>
<script src="templates/js/ifxdrop.js" type="text/javascript"></script>
<script src="templates/js/ifxblind.js" type="text/javascript"></script>
<script type="text/javascript" src="templates/js/imagebox new.js">
</script>
<ul title="My Gallery" id='Gallery'>
<?php
foreach($images as $k=>$v){
    echo  '<li><a  href="'.$folder.'/'.$v[2].'"  title="'.$k.'.'.$v[4].'"
    rel="imagebox-lights"><img src="'.$folder.'/thumbnail/'.$v[2].'.jpg"
    width="'.$width.'" height="'.$height.'"/></a></li>';
}
?>
<script type="text/javascript">
$(document).ready(
    function()
    {
        $.ImageBox.init(
            {
                loaderSRC: 'templates/images/loading.gif',
                closeHTML: '<img src="templates/images/close.jpg" />'
            }
        );
    }
);
</script>
<?php
    }
}
?>
```

【代码解读】

幻灯片模式显示图片方法的功能是显示开始和暂停按钮，单击图片时显示放大图像。该功能是通过切换在不同路径下的缩略图和原始图像，实现放大的切换效果的。

图片的滑动显示是通过在./js 文件夹下面定义了 3 个 Jquery 文件实现的图片拖动效果，单个图片单击放大显示（带有遮罩效果）方法类似，读者可以参照源文件中的 Jquery 文件。

图片网格列表显示是通过 foreach()函数，循环生成 HTML 页面标签生成对应的图片网格。

21.3.5　公用和配置文件

公用文件中包括了在项目中重复使用的代码，同时还包括了常用类的实例化操作。通过引用共用文件，可以节省很多重复的代码。

本节规划的公用文件代码中，有部分已经在前面多次出现，这类代码将不进行详细介绍，将主要介绍新添加的代码内容。公用文件中主要包括：

（1）用户认证。

（2）用户菜单显示。

（3）常用的部分函数。

公用文件的代码保存在根目录下，文件名为 global.php，详细代码如下所示。

```
<?php
//引用常用类
include 'class/mysql.class.php';
include 'class/table.class.php';
include 'class/cart.class.php';
$cart = new cart ();                              //实例化购物车类
$table = new table ();                            //实例化通用表格类
class globalClass extends mysql {
    function globalClass($host, $user, $pass, $db, $charset, $pre) {
        $this->mysql ( $host, $user, $pass, $db, $charset, $pre );
                                                  //链接数据库
    }
    //用户认证函数
    function auth(){
        $email = $_SESSION["i"]["email"];         //邮件账户 Session 变量
        $pass = $_SESSION["i"]["password"];       //登录密码 Session 变量
        $this->setSql("select id from #_members where email = '".$email."'
        and password = '".$pass."'");
        $this->query();
        if($this->getLines()<1){
            $this->alert('非法登录!','index.php');
        }
    }
    //登录用户信息
    function UserInfo(){
        $email = $_SESSION["i"]["email"];
        $pass = $_SESSION["i"]["password"];
        $this->setSql("select id,email,nickname,truename,sex,telphone,
        mobile,address from #_members where email = '".$email."' and
        password = '".$pass."'");
        $this->query();
        if($this->getLines()>=1){
            $user = $this->loadRow();
            echo '<li>登录邮箱：'.$user[1].'</li>';
            echo '<li>用户昵称：'.$user[2].'</li>';
            echo '<li>用户姓名：'.$user[3].'</li>';
            if($user[4]==1){
                echo '<li>性别：男</li>';
            }else{
                echo '<li>性别：女</li>';
            }
            echo '<li>电话：'.$user[5].'</li>';
            echo '<li>手机：'.$user[6].'</li>';
            echo '<li>地址：'.$user[7].'</li>';
        }else{
            $this->alert('非法登录!','index.php');
        }
    }
    //信息提示窗口
    function alert($info,$url,$time=3){
        echo $info;
        echo '<meta http-equiv="Refresh" content="'.$time.'; url='.$url.'"
        />';
        exit();
    }
    //登录信息显示
```

```
function loginStats() {
    if (isset ( $ SESSION ["i"] )) {
        switch ($ SESSION ["i"] ["admin"]) {
            case "0" :
                echo '
                    欢迎:' . $_SESSION ["i"] ["truename"] . '  
                       
                    <a class="navlogin" href="index.php">主页</a>
                    <a class="navlogin" href="product.php">产品列表</a>
                    <a class="navlogin" href="cart.php">购物车</a>
                    <a class="navlogin" href="logout.php">退出</a>
                ';
                break;
            case "1" :
                echo '
                    欢迎:' . $_SESSION ["i"] ["truename"] . '  
                       
                    <a class="navlogin" href="index.php">主页</a>
                    <a class="navlogin" href="product.php">产品列表</a>
                    <a class="navlogin" href="cart.php">购物车</a>
                    <a class="navlogin" href="user order.php">订单管理</a>
                    <a class="navlogin" href="user_product.php">产品管理</a>
                    <a class="navlogin" href="user.php">管理</a>
                    <a class="navlogin" href="logout.php">退出</a>
                ';
                break;
        }
    } else {
        echo '
        <a class="navlogin" href="index.php">主页</a>
        <a class="navlogin" href="product.php">产品列表</a>
        <a class="navlogin" href="cart.php">购物车</a>
        <a class="navlogin" href="login.php">登录</a>
        <a class="navreg" href="register.php">新用户注册</a>
        ';
    }
}
//推荐产品
function commendProduct() {
    $this->setSql ("select id,title,fileid from # product where commend
    = 'YES' order by id desc limit 0,10" );
    $this->query ();
    if ($this->getLines () > 0) {
        foreach ( $this->loadRowList () as $k => $v ) {
                                    //遍历结果集,生成记录列表
            echo '<LI id="' . $k . '"><a href="store.php?fileid=' . $v
            [2] . '">' . $v [1] . '</a></LI>';
        }
    } else {
        echo '<LI>暂无推荐产品</LI>';
    }
}
//最新产品
function newProduct() {
    $this->setSql ( "select id,title,fileid from # product order by id
    desc limit 0,10" );
    $this->query ();
    if ($this->getLines () > 0) {
```

```
            foreach ( $this->loadRowList () as $k => $v ) {
                                                    //遍历结果集,生成记录列表
                echo '<DIV class=user id="' . $k . '"><a href="store.
                php?fileid=' . $v [2] . '">' . $v [1] . '</a></DIV>';
            }
        } else {
            echo '<LI>暂无产品</LI>';
        }
    }
    //获取产品相关图片
    function aboutFiles($folder, $fileid) {
        include 'class/images.class.php'; //引用图片类
        $img = new images (); //实例化图片类
        $this->setSql ( "select id,fileid,filename,filetype,filetitle from
        # files where fileid = '" . $fileid . "'" );
        $this->query ();
        $st = $this->loadRowList ( 1 );
        $img->singleShow ( $folder, $st ); //使用 singleShow()函数显示图片
    }
    //购物车
    function getCart() {
        global $cart, $table;
        $table->_name = 'cartlist';
        $data = $cart->listArray ();
        if (count ( $data ) > 0) {
            $header = array ("编号", "名称", "数量", "价格" );
                                                //设置购物车列表的表头内容
            echo $table->normal ( $header, $data );
                                                //使用通用表格类，显示购物车内容
        } else {
            echo '<LI>暂无产品</LI>';
        }
    }

}
$g = new globalClass ( "localhost", "root", "password", "onlinestore",
"latin1", "o" );
?>
```

【代码解读】

（1）共用文件一开始就引用了 table 类和 cart 类，并进行了实例化，供以后的代码使用。

（2）loginStats()函数：根据会员的登录状态，显示不同的菜单内容。

（3）commend Product()函数：用于列出管理员推荐的产品。

（4）newProduct()函数：用于列出管理员最新添加的产品。

（5）aboutFiles()函数：根据参数列出与产品相关的图片，并使用 images 类中的 singleShow()函数显示数据。

（6）getCart()函数：读取购物车中的数据，并通过使用通用表格（table）类，以列表的形式显示出来。

21.4　会 员 管 理

购物网站的会员管理比较详细，包括会员的联系方式和住址等内容。虽然与会员相关

的字段有所增加，但是前面模块介绍过的与会员相关的操作基本相同。会员注册和登录的详细过程读者可以回顾前面的介绍，本节主要介绍会员管理过程上新增加的业务逻辑。

21.4.1　会员注册

会员注册与前面模块中介绍的设计模式有所不同，通常的用户（会员）之间的关系是平行的，用户之间没有制约关系。而在线购物平台模块，并不是所有的会员都可以增加产品信息、上传图片。所以在会员注册时，需要添加一些特殊的设置。

下面具体介绍会员注册部分的设计和实现步骤。

（1）技术要点：会员注册时，第一个会员将会被指定为管理员，在数据库中的表现是，用户表的 admin 字段的值被设置为 1。

（2）会员注册代码实现：会员注册页面保存在根目录下，文件名为 register.php。详细代码如下所示。

```
<?php
//用户注册代码
if(isset($_POST["action"]) and $_POST["action"]=="register"){
    $user = "";
    //获取 POST 变量中关于用户注册的信息
    $user["password"] = md5(trim($_POST["pass1"]));
    $user["email"] = trim($_POST["email"]);
    $user["nickname"] = trim($_POST["nickname"]);
    $user["truename"] = trim($_POST["truename"]);
    $user["telphone"] = trim($_POST["telphone"]);
    $user["mobile"] = trim($_POST["mobile"]);
    $user["address"] = trim($_POST["address"]);
    $user["sex"] = intval(trim($_POST["gender"]));
    $g->setSql("select id from #_members");
    $g->query();
    $lines = $g->getLines();
    if($lines>0){                                        //设置参数值
        $user["admin"] = 0;
    }else{
        $user["admin"] = 1;                              //管理员标识
    }
    $g->setSql("select id from #_members where email = '".$user ["email"].
    "'");//设置 SQL 语句
    $g->query();                                         //运行 SQL 语句
    if($g->getLines()>0){
        $g->alert('数据库存在相同的电子邮件地址，请尝试其他电子邮箱。','
        register.php');
    }else{
        $g->insertObject("#_members",$user);
        $g->alert('会员注册成功，请稍后...','index.php');
    }

}
?>
```

在首页单击“新用户注册”链接，或者在地址栏输入 http://localhost/b2c/register.php，运行后界面如图 21.7 所示。



图 21.7　会员注册

【代码解读】

（1）会员注册省略的部分代码是 HTML 表单及相关的界面代码，详细代码请查阅光盘内容。

（2）会员注册时，并没有提供用户名，而是使用邮箱作为登录时的登录凭证。

（3）在保存会员注册信息前，检查会员数据表中是否已经存在会员。如果会员数据表中已经存在记录，那么注册会员的 admin 的值为 0，即普通会员；当会员数据表中的记录为 0 时，注册会员的 admin 的值为 1，即管理员。

普通会员可以浏览产品列表，使用购物车等功能。而管理员不仅可以使用普通会员的功能，而且拥有订单管理、产品管理等功能。

这里需要注意的是，“邮箱”和“密码”项为必选项目。其中“邮箱”被默认为用户系统登录的账号名称，如果注册信息不全或者有非法字符，则弹出提示信息。界面效果如图 21.8 所示。



图 21.8　注册异常

21.4.2　会员登录

会员登录过程中，因为存在两种会员，即管理员和普通用户。为了区分普通会员和管理员，在允许管理员访问的页面需要添加特殊的认证代码。下面具体介绍会员登录部分的设计和实现步骤。

（1）技术要点：普通权限的会员登录后访问的页面，使用共用文件中的 auth()函数进行登录信息验证。只允许管理员访问的页面，需要检查登录信息中，是否含有管理员标记的内容。

（2）会员登录代码实现：会员登录页面保存在根目录下，文件名为 login.php。详细代码如下所示。

```
<?php
//用户登录代码
if (isset ( $_POST ["action"] ) and $_POST ["action"] == "login") {
    //查询数据表中的会员资料是否存在
    $g->setSql ( "select id,nickname,truename,admin from # members where
    email = '" . trim ( $_POST ["email"] ) . "' and password = '" . md5 ( trim
    ( $_POST ["password"] ) ) . "'" );
    $g->query ();
    if ($g->getLines () > 0) {              //当结果集记录数大于 0 时，登录
        $user = $g->loadRow ();
        //将会员登录信息保存到 SESSION 中
        $_SESSION ["i"] ["id"] = $user [0];
        $_SESSION ["i"] ["nickname"] = $user [1];
        $_SESSION ["i"] ["truename"] = $user [2];
        $_SESSION ["i"] ["admin"] = $user [3];
        $_SESSION ["i"] ["email"] = trim ( $_POST ["email"] );
        $_SESSION ["i"] ["password"] = md5 ( trim ( $_POST ["password"] ) );
        $g->alert ( '用户登录成功，请稍候...', 'user.php' );
                                        //跳转到会员资料管理页
    } else {
        echo '<font color="red">无效的登录信息，请重试！</font>';
    }
}
?>
```

【代码解读】

用户输入邮箱和登录密码后，提交到用户登录代码。用户登录信息在通过验证后，在 SESSION 数据中记录会员的昵称、姓名、邮箱、密码、权限等信息。权限信息是保存在 admin 变量中的，当变量值为 1 时，表示当前用户拥有管理员权限；当变量值为 0 时，表示当前用户是普通用户。

在首页单击“登录”链接，或者在地址栏输入 http://localhost/b2c/login.php 运行后，界面如图 21.9 所示。

21.4.3　会员资料管理界面

会员资料管理界面中，根据会员数据表中的字段，以表单的形式，列出了可以修改的会员资料的内容。下面具体介绍会员资料管理界面的设计和实现步骤。



图 21.9 会员登录

（1）技术要点：会员资料界面的修改，核心功能就是读取数据库内容、创建表单、提交表单的过程。在这个过程中，需要注意表单提交元素显示的内容，要与会员资料表中的内容一致。

（2）会员资料管理界面：会员资料管理界面的代码保存在根目录下，文件名为 user.php。详细代码如下所示。

```
<?php
//用户资源修改代码
if (isset ( $_POST ["action"] ) and $_POST ["action"] == "editPassword")
{ //检测用户是否提交约定变量
    //将用户提交的 POST 变量元素，保存到数组中
    $user ["email"] = $_SESSION ["i"] ["email"];
    $user ["password"] = md5 ( trim ( $_POST ["pass1"] ) );
    $user ["nickname"] = trim ( $_POST ["nickname"] );
    $user ["truename"] = trim ( $_POST ["truename"] );
    $user ["telphone"] = trim ( $_POST ["telphone"] );
    $user ["mobile"] = trim ( $_POST ["mobile"] );
    $user ["address"] = trim ( $_POST ["address"] );
    $user ["sex"] = intval ( trim ( $_POST ["gender"] ) );
    $g->updateObject ( "#_members", $user, "email" );
                                    //使用 updateObject()方法更新记录
    unset ( $_SESSION ["i"] );
    $g->alert ( '用户资料修改成功，请重新登录...', 'index.php' );
} else {
    $uid = $_SESSION ["i"] ["id"];      //获取 SESSION 保存的用户 ID
    $g->setSql ( "select nickname,truename,sex,telphone,mobile,address
    from #_members where id = " . $uid );
    $u = NULL;
    $g->loadObject ( $u );              //以对象方式加载会员数据
}
?>
```

【代码解读】

会员登录后，首先显示会员资料修改页面。user.php 文件中，该文件用于显示会员资料的表单代码，具体内容请查看源程序文件。本小节主要介绍用于保存和读取会员资料的 PHP 代码。

（1）读取：读取会员资料的代码，主要是使用已经定义好的数据库操作类中的方法。在读取会员资料后，为了方便使用，使用了数据库操作类中的 loadObjecct()函数，将返回的会员资料转化为对象。

（2）保存：在保存会员资料数据时，首先要对提交的数据进行校验，校验通过后，将

表单中的数据，保存在数组中。然后使用数据库操作类中的 insertObject()函数，将数组插入到指定的表中，完成数据保存的操作。这个过程中并没有涉及 SQL 语句，保存过程中的 SQL 语句生成，运行 SQL 语句等操作，都是由 insertObject()函数完成的。

会员登录后，跳转到会员资料管理界面，界面如图 21.10 所示。



图 21.10　会员资料管理界面

由于使用者有误操作的情况出现，因此程序在设计上，首先要验证两次输入的密码是否相同，以及是否有非法字符输入，这是必要的验证功能。当密码输入异常时，提示错误信息，同时终止程序运行保证修改的正确性。界面效果如图 21.11 所示。



图 21.11　会员密码修改

21.5　产 品 管 理

本节主要演示通用表格类、文件上传类等代码的使用方法。通过本节的内容，可以验证通用表格的复用度、可用性。

21.5.1　产品信息管理

产品管理界面，主要是列出购物网站已经有的产品内容，在产品管理界面里，实现了可以添加、删除产品等功能。下面具体介绍产品信息管理的设计和实现步骤。

（1）技术要点：产品管理中添加产品的代码很简单，就不做过多介绍，详细代码请查阅光盘内容。产品管理中的产品列表实现是本节介绍的重点内容。通过本节的介绍，可以了解通用表格类的使用方法。

（2）产品管理实例代码实现：产品管理页面保存在根目录下，文件名为 user_product.php。详细代码如下所示。

```
<?php
session_start ();
include 'global.php';
$g->auth ();
$uid = $_SESSION ["i"] ["id"];           //获取 SESSION 中关于登录用户的信息
if ($_SESSION ["i"] ["admin"] == 0) {
    $g->alert ( "无权访问该页!", "index.php" );
}
?>
........//参考源文件
<?php
//表单处理代码
if (isset ( $_POST ["do"] )) {
    $action = strval ( $_POST ["do"] );     //获取用户提交的约定变量
    $error = "";
    switch ($action) {
        case "addProduct" :         //当约定变量值为 addProduct 时,运行以下代码
            $f = $_POST;
            if ($f ["title"] == "") {
                $error .= "产品名称不能为空<br>";
            }
            $g->setSql ( "select id from #_product where title = '" . $f
            itle"] . "'" );
            $g->query ();
            if ($g->getLines () > 0) { //获取结果记录数
                $error .= "产品已经存在<br>";
            }
            if ($error == "") {          //当没有错误信息时，继续运行
                /*****创建文件夹代码开始****/
                $path = $pathSign = "/";
                $path = "folder/" . $uid . "/" . date ( "Y" ) . "/" . date
                 ( "m" ) . "/" . date ( "d" ); //创建保存路径
                $dirArray = explode ( $pathSign, $path . "/thumbnail" );
                                        //切分文件保存路径
```

```
                $tempDir = '';
                foreach ( $dirArray as $dir ) {
                    $tempDir .= $dir . $pathSign;
                    $isFile = file_exists ( $tempDir );//检测文件夹是否存在
                    clearstatcache ();                  //清除缓存记录
                    if (! $isFile && ! is dir ( $tempDir )) {
                        @mkdir ( $tempDir, 0777 );
                                                    //当文件夹不存在时，创建文件夹
                    }
                }
                /******创建文件夹代码结束***/
                $p ["title"] = $f ["title"];//将用户提交变量，保存到新数组中
                $p ["intro"] = $f ["intro"];//将用户提交变量，保存到新数组中
                $p ["price"] = $f ["price"];
                $p ["mprice"] = $f ["mprice"];
                $p ["fileid"] = md5 ( time () . $uid );
                $p ["commend"] = $f ["commend"];
                $p ["images"] = $path;      //获取图片文件保存路径
                $g->insertObject ( "# product", $p );//向数据库中插入新记录
                $g->alert ( '添加产品成功', 'user_product.php', 1 );
                                            //向数据表中添加新产品
            } else {
                $g->alert ( $error, 'user product.php' );
            }
            break;
        case "delete" :
            if (count ( $ POST ["cid"] ) > 0) {
                foreach ( $ POST ["cid"] as $k => $v ) {
                    $g->setSql ( "delete from #_product where id = '" . $v .
                    "'" );
                    $g->query (); //运行删除产品的 SQL 语句
                }
                echo '删除成功';
            } else {
                echo '请选择要删除的记录';
            }
            break;
        case "commend" :
            if (count ( $ POST ["cid"] ) > 0) {
                foreach ( $_POST ["cid"] as $k => $v ) {
                                            //遍历用户提交的表单变量
                    $g->setSql ( "update #_product set commend = 'YES' where
                    id = '" . $v . "'" );
                    $g->query ();           //运行修改推荐状态的 SQL 语句
                }
                echo '推荐成功';
            } else {
                echo '请选择要推荐的记录';
            }
            break;
        case "upload" :
            if (count ( $ POST ["cid"] ) > 0) {
                $upid = end ( $ POST ["cid"] );
                                        //获取用户提交表单变量的最后一个元素
                //从产品数据库中，检索出 fileid，用于读取与产品相关的图片文件
                $g->setSql ( "select fileid from # product where id = '" .
                $upid . "'" );
                $g->query ();
```

```
                $up = $g->loadRow ();       //获取记录数组
                //创建链接到图片上传的页面
                echo '<script>window.location.href="user_product_upload.
                php? upid=' . $up [0] . '";</script>';
            } else {
                echo '请选择要上传图片的记录';
            }
            break;
    }
}
?>
<!---------产品列表开始--------->
<?php
$header = array ("ID", "产品名称", "市场价", "商城价", "推荐" );
                                                //定义用于表格的标头数据
$g->setSql ( "select id,title,price,mprice,commend from #_product" );
$g->query ();
$data = $g->loadRowList ( 1 ); //取得查询的产品结果集
$toolbar = array ("delete" => array ("value" => "删除", "action" => "" ),
"commend" => array ("value" => "推荐", "action" => "" ), "upload" => array
("value" => "上传图片", "action" => "user product upload.php" ) );
                                                //定义用于表格的工具栏数据
echo $table->normal ( $header, $data, true, $toolbar );
                                                //使用 normal()方法，显示产品
?>
```

【代码解读】

user_product.php 中的代码包括 3 部分。

（1）会员认证：在 user_product.php 一开始，先进行了会员认证操作，防止非注册用户访问本面。同时检查用户登录信息中 admin 变量的值，当 admin 变量的值不为 0 时，才允许访问本页。

（2）表单处理代码：用于处理本页提交的 POST 变量，根据$_POST["do"]变量的内容，来执行相应的代码。

（3）产品列表：使用通用表格类列出已经添加的产品信息。下面介绍通用表格类的详细使用方法。

- 表头参数：其变量类型是一个一维数组，内容是与 SQL 语句列表中对应的查询字段。表头参数第一个值一般是 ID，与数据表中的字段 id 绑定后，形成表单中的复选框控件。
- 数据列表：以二维数组的形式保存产品数据，通过通用表格类中兼容性的设置，可以接收以数字为索引值的二维数组、以字符为索引值的二维数组，以及相关的对象类型的数据。
- 表单控制：当表单控制的参数值为 true 时，表格内的复选框才能显示出来。
- 工具栏：以二维数组保存的工具栏信息，第一维数组的索引值是表单中名称为 do 的控件的值。与其对应的一维数组中，包括两个元素，第一个元素是以 value 为索引值的元素，其值是工具栏按钮的名称。第二个元素是以 action 为索引值的元素，其值是表单 action 属性的值。
- 表格宽度：默认为 100%，本节中已经省略，使用其默认值。

在首页单击“产品列表”链接，或者在浏览器中输入 http://localhost/b2c/user_product.

php，界面效果如图 21.12 所示。

下面添加一个产品信息。以添加一个数码相机为例，必要的添加信息为市场价、商城价，可选项为简介和推荐复选框。填写信息的界面如图 21.13 所示。

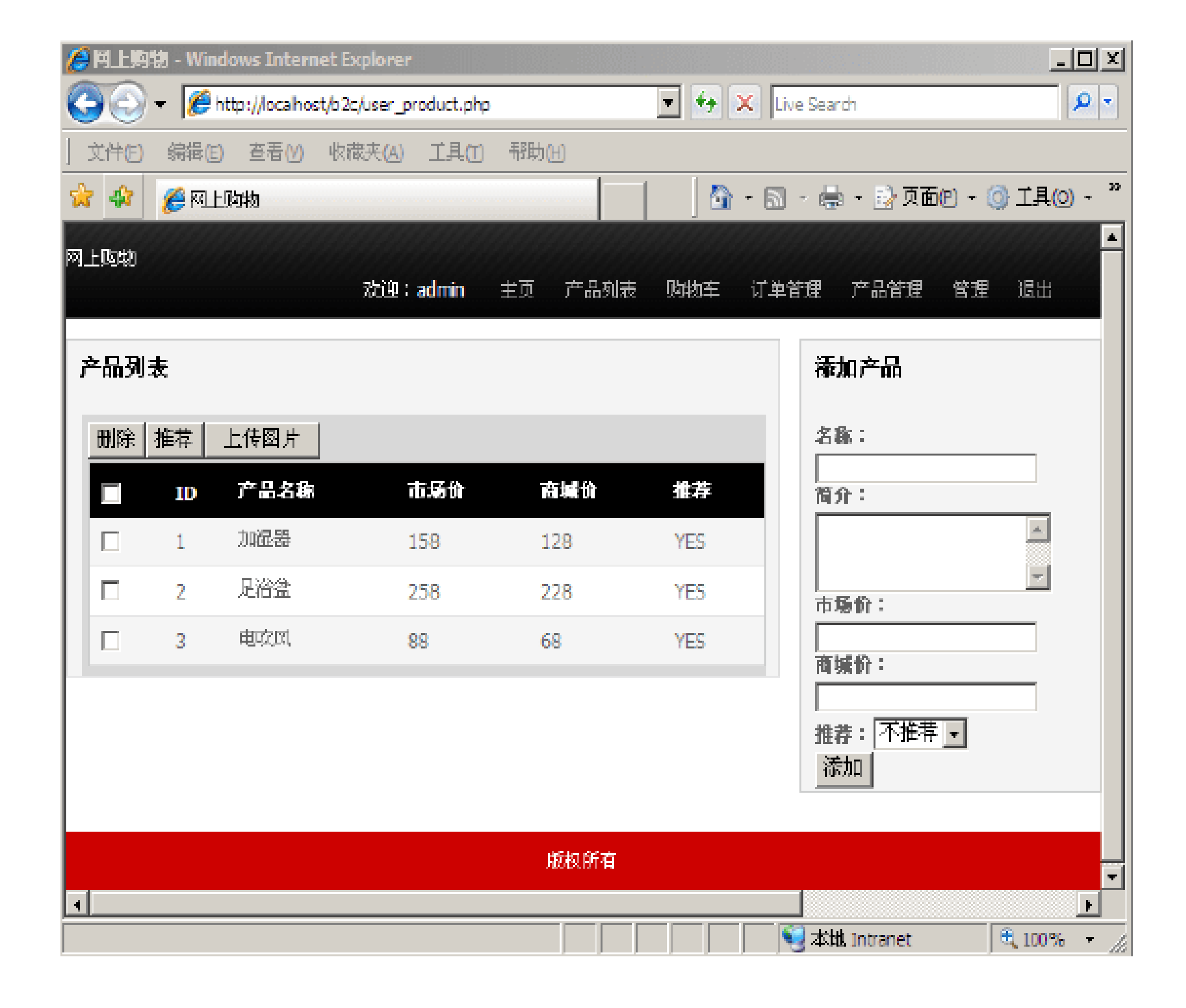


图 21.12　产品管理界面

添加产品

名称：

数码相机

简介：

时尚、轻薄成像清晰。

市场价：

1099

商城价：

999

推荐：不推荐

添加

图 21.13　添加产品信息

添加产品信息后提示安装成功信息，跳转后产品列表页面在最下面出现。产品列表信息界面如图 21.14 所示。

产品列表

删除　推荐　上传图片

	ID	产品名称	市场价	商城价	推荐
□	1	加湿器	158	128	YES
□	2	足浴盆	258	228	YES
□	3	电吹风	88	68	YES
□	4	数码相机	1099	999	NO

图 21.14　新添加商品后列表

在最新添加的产品信息列表选项前勾选，单击“推荐”按钮，将最新添加的数码相机列为推荐产品。推荐后产品列表如图 21.15 所示。

产品列表

推荐成功

删除　推荐　上传图片

	ID	产品名称	市场价	商城价	推荐
□	1	加湿器	158	128	YES
□	2	足浴盆	258	228	YES
□	3	电吹风	88	68	YES
□	4	数码相机	1099	999	YES

图 21.15　新产品推荐

21.5.2　产品图片管理

产品图片管理实现的功能，主要是将与产品相关的图片上传到服务器上，并与产品建立索引，方便在查看产品时，访问相关的图片。

下面具体介绍产品图片管理的设计和实现步骤。

（1）技术要点：产品图片管理过程中，上传的代码主要使用到了已经定义好的上传操作类 upload.class.php 中的代码。

在创建产品后，勾选要管理的产品，单击相应的按钮，来完成管理动作。其中，删除和推荐操作都是在本页完成的，而上传图片操作是在产品图片上传页中完成的，

（2）产品图片管理实例代码：产品图片上传页保存在根目录下，文件名为 user_product_upload.php。详细代码如下所示。

```
<?php
session_start ();
include 'global.php';
$g->auth ();
?>
........//参考程序源代码
<?php
if (isset ( $_POST ["do"] ) && $_POST ["do"] == "upload") {
    $upid = end ( $_POST ["cid"] );              //获取用户提交的约定变量
    $_SESSION ["temp"] ["upid"] = $upid;         //保存到 SESSION 变量中
} elseif (isset ( $_SESSION ["temp"] ["upid"] )) {
    $upid = $_SESSION ["temp"] ["upid"];         //从 SESSION 中获取约定变量
}
$g->setSql ( "select id,title,intro,images,fileid from #_product where id
= '" . $upid . "'" );
$f = NULL;
$g->loadObject ( $f );                           //以对象方式加载产品信息
//表单处理代码
if (isset ( $_POST ["do"] )) {
    $error = "";
    switch ($_POST ["do"]) {                     //判断用户提交的约定变量
        case "addFiles" :                //当约定变量值为 addFiles 时，上传文件
            include_once ("class/upload.class.php");   //引用文件上传类
            $u = new upload ();                  //实例化文件上传类
            $path = $f->images . "/";
            $p = $_POST;
            $up = $u->uploadFile ( "file", 1, $path );
                                                 //使用上传文件类上传文件
            if ($up ["filestat"] == "false") {
                $error .= $up ["filename"];
            }
            if ($error == "") {          //当上传没有错误时，保存上传文件信息
                $file ["fileid"] = $f->fileid;
                $file ["filetitle"] = $p ["title"];
                $file ["filename"] = $up ["filename"];
                $file ["filetype"] = $up ["filetype"];
                $g->insertObject ( "#_files", $file );
                                         //将文件上传后的信息保存到文件数据表中
            } else {
                $g->alert ( $error, 'user_product_upload.php' );
```

```
            }
            break;
        case "delete" :
            if (count ( $_POST ["cid"] ) > 0) {
                foreach ( $_POST ["cid"] as $k => $v ) {
                                        //遍历用户提交的表单变量
                    $g->setSql ( "delete from # files where id = '" . $v .
                    "'" );
                    $g->query (); //运行删除文件的 SQL 语句
                }
            } else {
                echo '请选择要删除的记录';
            }
            break;
    }
}
?>
........//参考程序源代码
<!---------产品列表开始--------->
<?php
$header = array ("ID", "名称" );            //设置通用表格类需要的表头信息
$g->setSql ( "select id,filetitle from # files where fileid = '" . $f->fileid .
"'" );
$g->query ();
$data = $g->loadRowList ( 1 );              //获取数据库中的记录
$toolbar = array ("delete" => array ("value" => "删除", "action" => "" ) );
                                            //设置通用表格类的工具栏数据
echo $table->normal ( $header, $data, true, $toolbar );
                                            //使用通用表格类显示已经上传的文件
?>
<!---------结束--------->
........//参考程序源代码
```

【代码解读】

user_product_upload.php 中的内容，主要也是介绍了通用表格类的使用，同时还复用了在核心程序类中介绍过的文件上传类的文件上传代码。

在首页单击“产品列表”链接，或者在浏览器中输入 http://localhost/b2c/user_product_upload.php，界面效果如图 21.16 所示。



图 21.16　产品图片上传管理

21.6　产品展示与购物车

在管理员增加完产品后，普通会员就可以从网站购买产品了。购买产品的过程中要使用到购物车功能，本节主要介绍购物车类的使用方法。

21.6.1　购物车

要购买产品，首先需要显示出产品，并提供添加到购物车的链接。本节将实现产品列表和购物车列表内容的显示。

首页产品列表显示，使用的是普通的列表方式。而购物车列表，是使用通用表格类完成的内容显示。

下面具体介绍与产品相关的购物车的设计和实现步骤。

（1）技术要点：购物车类已经在功能规划时，完成了购物车类的编码，本节主要是对其进行应用操作。向购物车中添加记录分为单条添加和多条添加。

- 单条添加：在浏览产品详细内容时，单击加入购物车链接实现的。
- 多条添加：在产品列表中，使用多选通过表单提交后添加的。

（2）实例代码：首页内容保存在根目录下，文件名为 index.php。详细代码如下所示。

```
<?php
session start();
include 'global.php';       //加载配置文件
<HTML xmlns="http://www.w3.org/1999/xhtml">
<HEAD>
<TITLE>网上购物</TITLE>
</HEAD>
<BODY>
<DIV id=navwrap class=navwrap>
    <div class="nav">
    <div class="navinner"><br>
        网上购物
            <div class="navsearch" style="left: 735px">
                <form action="search.php" method="post" target=" blank">
                <div class="input">
                    <input type="hidden" name="type" value="file" />
                    <input class="input" name="keyword" type="text"
                    onFocus="this.className='input2'"value="搜产品" onClick=
                    "this.value='';"    onblur="this.value = this.value
                    =='' ? '资源名称' : this.value;this.className='input'" />
                </div>
                <input class="but"  type="submit" value=" " /></form>
            </div>
            <div class="other" style="left:200px">
            <?php $g->loginStats(); //登录模块?>
            </div>
</div>
    </div>
</DIV>
<DIV class=wrap>
```

```
    <DIV class=left>
        <DIV class=commend>
            <DIV class=group>
                <DIV class=title>最新产品</DIV>
                <?php $g->newProduct();     //获得产品列表?>
            </DIV>
            <DIV class=space>
                <DIV class=title>购物车</DIV>
                    <UL class=cool>
                <?php $g->getCart();        //获得购物车中内容?>
                    </UL>
                </DIV>
            </DIV>
        </DIV>
        <DIV class=right>
            <DIV class=play>
                <DIV class=title>推荐产品</DIV>
                    <DIV class=playwrap>
                        <UL id=scrollPlay>
                        <?php $g->commendProduct();?>
                        </UL>
                    </DIV>
                </DIV>
            </DIV>
        </DIV>
    <DIV class=clear></DIV>
    </DIV>
</DIV>
</BODY>
</HTML>
```

【代码解读】

index.php 中的代码包括 3 部分：

（1）最新产品：使用共用文件中的 newProduct()函数，列出最新的 10 项产品。

（2）购物车：使用共用文件中的 getCart()函数，实现无表单控件，无工具栏的购物车内容列表。

（3）推荐产品：使用共用文件中的 commendProduct()函数，实现推荐产品列表。

在浏览器中输入 http://localhost/b2c/index.php，界面效果如图 21.17 所示。

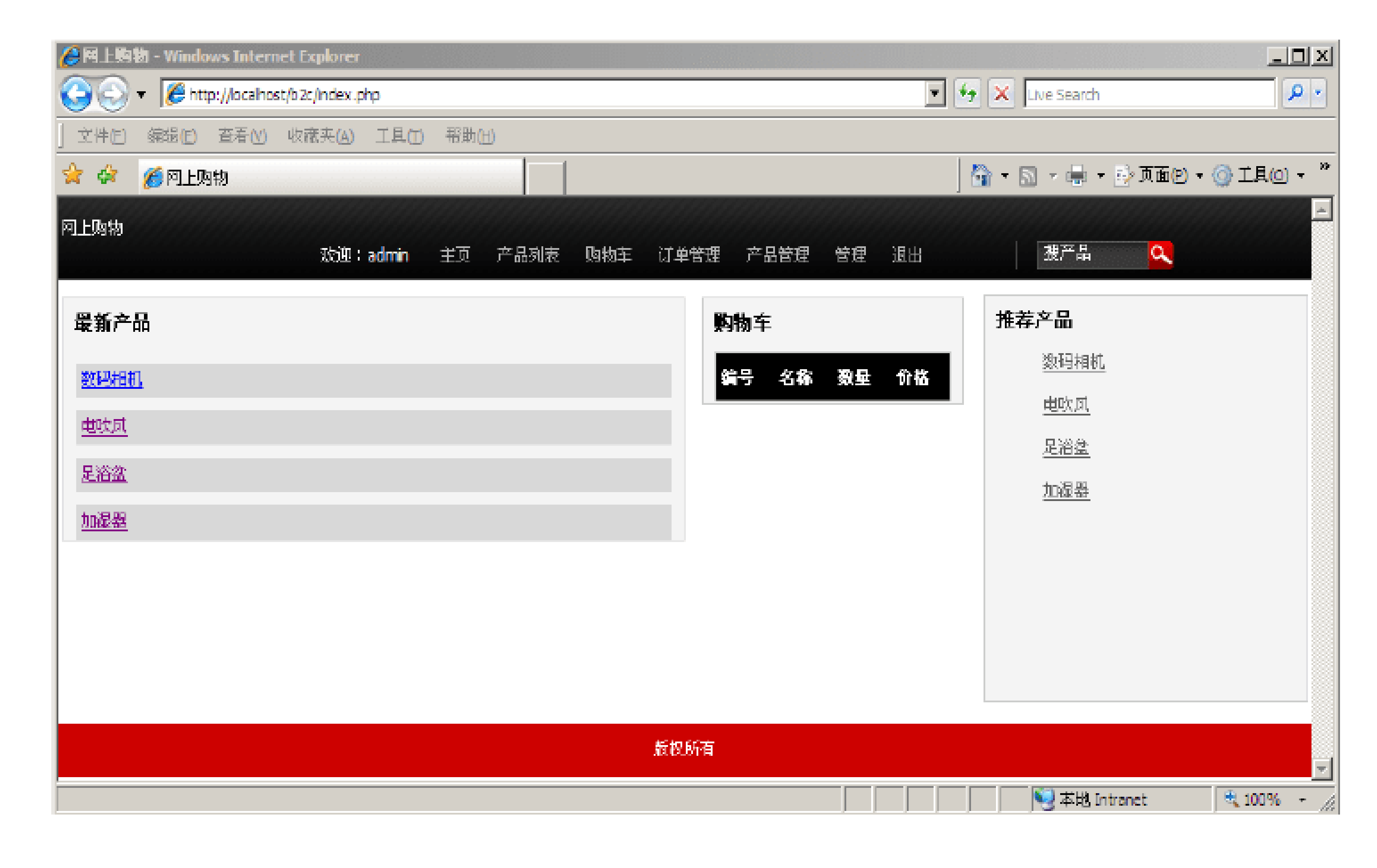


图 21.17　在线购物平台首页

21.6.2　产品详细页实现

产品详细显示页面，需要显示产品的名称、简介、价格等信息。同时需要把与产品相关的图片列出来，并增加放大显示的效果。在产品详细页面中，还需要有把产品添加到购物车的链接。下面具体介绍产品详细显示页面的设计和实现步骤。

（1）技术要点：与产品相关图片的显示，使用了已经定义好的 images.class.php 文件中的代码。详细代码请参照光盘中相关文件。

把产品添加到购物车的链接，以 GET 方式向购物车页提交一个值。购物车页面会根据这个值，读取产品相关的信息，并将产品添加到购物车中。

（2）产品详细显示：单击首页最新产品栏目中的任意一个链接，进入产品详细显示页面，产品详细显示页面保存在根目录下，文件名为 store.php。详细代码如下所示。

```
<?php
session_start();
include 'global.php';
?>
........//参考随书光盘
<?php
if(isset($_GET["fileid"])){
    $fileid = strval($_GET["fileid"]); //获取链接中的约定变量
    $g->setSql("select title,intro,price,mprice,fileid,commend,images
    from #_product where fileid = '".$fileid."'");
    $f = NULL;
    $g->loadObject($f);                //以对象形式，加载产品信息
}elseif(isset($_POST["do"]) and isset($_POST["cid"]) and count($_POST
["cid"])>0){
    $fileid = end($_POST["cid"]);      //获取表单中的约定变量
    $g->setSql("select title,intro,price,mprice,fileid,commend,images
    from #_product where id = '".$fileid."'");
    $f = NULL;
    $g->loadObject($f);                //以对象形式加载产品信息
}
?>
    ........//参考随书光盘
    <DIV class=title><?php echo $f->title;?></DIV>
    <DIV class=user>
        简介：<?php echo $f->intro;?><br>
    </DIV>
    <DIV class=user>
        市场价：<?php echo $f->price;?><br>
    </DIV>
    <DIV class=user>
        商城价：<?php echo $f->mprice;?><br>
    </DIV>
    <DIV class=user>
        推荐商品：<?php echo $f->commed;?><br>
    </DIV>
    <DIV class=user>
        <a href='cart.php?id=<?php echo $f->fileid; ?>'><strong>放入购物车
        </strong></a>
        <?php echo$g->aboutFiles($f->images,$f->fileid); ?>
    </DIV>
```

```
</DIV>
........//参考随书光盘
```

在浏览器中输入 http://localhost/b2c/store.php 运行后，界面效果如图 21.18 所示。



图 21.18　产品详细页面

注意：细心的读者可以发现与别的页面不同的地方，浏览器右下角出现了“页面上有错误”的提示信息。这是由于同一个页面出现了二次引用 jQuery 框架的原因。所以在规划同一个项目的功能时，对于引用文件，一定要进行统一的处理，以防止这类错误的发生。

【代码解读】

store.php 可以处理 GET 和 POST 两种方法传递的数据，以完成产品详细内容的显示。以 GET 方法传递的数据，以从首页最新产品列表链接中产生的；而以 POST 方式产生的数据，是在产品列表中的表单产生的。

单击产品详细页面中的“放入购物车”链接，产品将添加到购物车中，界面结果如图 21.19 所示。



图 21.19　购物车详细页面

以上是以单条添加产品到购物车的方法。下面演示向购物车中同时添加多条记录的代码，单击“产品列表”链接，进入产品列表页面。

21.6.3　产品列表

首页上的产品列表，只是显示了部分产品，而产品列表则显示了购物网站中所有的产品内容。通过为产品列表添加相关的代码，可以实现一次添加多个产品到购物车的功能。下面具体介绍产品列表的设计和实现步骤。

（1）技术要点：产品列表是使用通用表格类实现的，在创建产品列表时，需要考虑到列表能使用到的功能。

（2）将产品放入购物车：通过多选产品记录前边的复选框，并将这些记录提交给购物车页，购物车页根据提交的数据，把相关产品批量添加到购物车中。

（3）查看详情：选择产品记录，将表单数据提交到产品详细显示页面，显示产品详细信息。

（4）产品列表：产品列表页面保存在根目录下，文件名为 product.php。详细代码如下所示。

```
<?php
session_start();            //初始化 SESSION 数据
include 'global.php';       //引用公用文件
?>
<HEAD>
<TITLE>网上购物</TITLE>
</HEAD>
<BODY>
<DIV id=navwrap class=navwrap>
    <div class="nav">
    <div class="navinner"><br>
        网上购物
            <div class="navsearch" style="left: 735px">
                <form action="search.php" method="post" name="search"
                target="_blank">
                <div class="input">
                    <input type="hidden" name="type" value="file" />
                    <input class="input" name="keyword" type="text"
                    onFocus="this.className='input2'"value="搜产品" onClick=
                    "this.value='';"onblur="this.value = this.value ==='' ?
                    '资源名称' : this.value;this.className='input'" />
                </div>
                <input class="but" name="searchButton" type="submit"
                value=" " /></form>
            </div>
            <div class="other" style="left:200px">
            <?php $g->loginStats();?>
            </div>
    </div>
    </div>
</DIV>
<DIV class=wrap>
    <DIV class=left>
        <DIV class=commend>
            <DIV class=group>
```

```
                <DIV class=title>最新产品</DIV>
                <DIV class=user style="text-decoration: underline;">
    <?php
    $g->setSql("select id,title,price,mprice,commend from # product order
    by id desc");
    $g->query();
    if($g->getLines()>0){
        $header = array("编号","名称","市场价","商城价","推荐",);
                                                //设置表格类使用的参数
        $data = $g->loadRowList(1);             //获取产品数据数组
        $toolbar = array(
            "cart"   =>array("value"=>"放入购物车","action"=>"cart.php"),
            "detail" =>array("value"=>"查看详情","action"=>"store.php")
        );
        //设置通用表格类工具栏
        $table-> name ="productTable";
        echo $table->normal($header,$data,true,$toolbar);//显示产品列表
    }else{
        echo '<LI>暂无产品</LI>';
    }
    ?>
    </DIV>
            </DIV>
            <DIV class=space>
                <DIV class=title>购物车</DIV>
                    <UL class=cool>
                    <?php $g->getCart();?>
                    </UL>
                </DIV>
            </DIV>
        </DIV>
    <DIV class=right>
            <DIV class=play>
                <DIV class=title>推荐产品</DIV>
                    <DIV class=playwrap>
                        <UL id=scrollPlay>
                        <?php $g->commendProduct();?>
                        </UL>
                    </DIV>
                </DIV>
            </DIV>
        </DIV>

        <DIV class=clear></DIV>
    </DIV>
</DIV>
```

单击导航栏中的“产品列表”链接，或者在浏览器中输入 http://localhost/b2c/product.php，运行后界面结果如图 21.20 所示。

【代码解读】

product.php 以列表的形式，列出了购物网站中所有的产品。其主要代码是使用通用列表类，创建了一个表格和一个工具栏。

表格的内容是购物网站中所有的产品。在对大量数据进行列表时，往往会涉及分页。本节提到的通用列表类，并没有实现数据分页的功能，感兴趣的读者，可以在通用列表类的基础上，实现分页管理。



图 21.20　产品列表页面

工具栏中包括两个按钮，选择列表中的一条记录，单击“查看详情”按钮，会将当前表单的内容，提交到 store.php 文件中，由 store.php 文件中的代码处理相关数据。

单击产品列表工具栏下方的选择框，全选产品列表中的记录，再单击“放入购物车”按钮，将当前表单的数据提交到 cart.php 文件中。这时，购物车列表页中的结果如图 21.21 所示。



图 21.21　购物车详细页面

产品列表页向购物车列表页，提供了多条产品数据，购物车列表页中的代码对这些数据进行处理后，添加到购物车列表中。购物车列表页保存在根目录下，文件名为 cart.php，详细代码如下所示。

```
<?php
session start ();
include 'global.php';
//获取以 GET 方法传递到本页的数据
if (isset ( $ GET ["id"] )) {        //获取通过链接提交的约定变量
    $g->setSql ( "select id,title,mprice from #_product where fileid = '" .
    strval ( $ GET ["id"] ) . "'" );
    $p = NULL;
```

```
    $g->loadObject ( $p );          //以对象方式加载产品数据
    $cart->addItem ( $p->id, $p->title, 1, $p->mprice );
                                    //运行购物车类的 additem()添加一条记录
}
//获取以 POST 方法传递到本页的数据
if (isset ( $_POST ["do"] )) {
    switch ($_POST ["do"]) {        //判断用户提交的约定变量
        case "cart" :           //当约定变量等于 cart 时，向购物车中批量添加记录
            if (count ( $_POST ["cid"] ) > 0) {
                foreach ( $_POST ["cid"] as $v ) {
                    $g->setSql ( "select id,title,mprice from #_product
                    where id = '" . $v . "'" );
                    $p = NULL;
                    $g->loadObject ( $p );          //以对象方式加载产品信息
                    $cart->addItem ( $p->id, $p->title, 1, $p->mprice );
                                    //添加购物车记录
                }
            }
            break;
        case "delete" :
            if (count ( $_POST ["cid"] ) > 0) {
                foreach ( $_POST ["cid"] as $v ) { //遍历表单数据
                    $cart->removeItem ( $v );       //删除购物车记录
                }
            }
            break;
        case "order" : //当约定变量等于 order 时
            if (isset ( $_SESSION ["i"] )) {
                $cartArray = $cart->listArray (); //获取购物车记录的数组
                if (count ( $_POST ["cid"] ) > 0) {
                    $total = 0;
                    foreach ( $_POST ["cid"] as $id ) {
                                                    //遍历用户提交的表单变量
                        $order = "";
                        $order ["pid"] = $cartArray [$id] ["id"];
                        $order ["mid"] = $_SESSION ["i"] ["id"];
                        $order ["price"] = $cartArray [$id] ["price"];
                        $order ["number"] = $cartArray [$id] ["number"];
                        $sprice = $cartArray [$id] ["price"] * $cartArray
                        [$id] ["number"];
                        $order ["total"] = $sprice;
                        $g->insertObject ( "#_order", $order );
                                                    //添加新的定单记录
                        $total += $sprice;
                    }
                    $cart->clear ();                //清空购物车数据
                    //显示结算数据，并返回首页
                    $g->alert ( "结算成功，共计费用：" . $total . "元",
                    "index.php", 5 );
                } else {
                    $g->alert ( "请选择确定购买的产品", "cart.php", 1 );
                }
            } else {
                $g->alert ( "注册用户请 <a href='login.php'>登录</a> 后再进行
                结算，非注册用户请 <a href='register.php'>注册</a> 后再登录。",
                "login.php", 5 );
            }
            break;
```

```
    }
}
?>
........//参考随书光盘
        <?php
        $data = $cart->listArray ();
        if (count ( $data ) > 0) {
            $header = array ("编号", "名称", "数量", "价格" );
                                        //设置通用表格类表头
            $toolbar = array ("delete" => array ("value" => "删除选择的物品
            ", "action" => "" ), "order" => array ("value" => "结算", "action"
            => "" ) );                  //设置通用表格类工具栏
            echo $table->normal ( $header, $data, true, $toolbar );
                                        //显示购物车中的记录
            echo "共计: " . $cart->_total () . "元";
        } else {
            echo '<LI>暂无产品</LI>';
        }
        ?>
........//参考随书光盘
```

【代码解读】

cart.php 中的代码主要包括 3 部分：

（1）接收以 GET 方法传递的数据，向购物车添加一条数据。

（2）接收以 POST 方法传递的数据，并根据$_POST["do"]变量的值，完成相关操作。当$_POST["do"]的值为 cart 时，向购物车中添加在产品列表页中选择的多条数据；当$_POST["do"]的值为 delete 时，根据当前页提交的 POST 数据，删除相应的购物车数据；当$_POST["do"]的值为 order 时，对选择的产品进行结算，并保存到订单数据表 o_order 中。由于只有注册用户才能进行结算，所以在结算过程中，应该先检查用户是否已经登录，再进行下一步的操作。

（3）使用通用列表类，列出购物车已经存在的数据，并创建两个按钮。

21.7 订 单 管 理

当会员对购物车中的产品进行结算后，会产生订单数据。管理员登录后，单击“订单管理”链接，进行订单管理页面管理订单。下面具体介绍订单管理的设计和实现步骤。

（1）技术要点：订单管理主要演示的是通用列表类的使用方法，通过使用通用列表类，列出订单数据表中已经存在的数据，并为订单列表创建“完成发货”的工具栏按钮。根据工具栏按钮，创建相应的 PHP 代码，完成订单状态的修改。

（2）订单管理实例代码：订单管理页面保存在根目录下，文件名为 user_order.php。详细代码如下所示。

```
<?php
session_start ();
include 'global.php';
$g->auth ();
$uid = $_SESSION ["i"] ["id"];
if ($_SESSION ["i"] ["admin"] == 0) {       //检测用户是否有权限访问本页
```

```
    $g->alert ( "无权访问该页!", "index.php" );
}
?>
<HEAD>
<TITLE>网上购物</TITLE>
</HEAD>
<BODY>
<DIV id=navwrap class=navwrap>
<div class="nav">
<div class="navinner"><br>网上购物
<div class="navsearch" style="left: 735px">
<form action="search.php" method="post" target=" blank" name="search"id=
"search">
<div class="input">
    <input type="hidden" name="type" value="file" />
    <input class="input" name="keyword" type="text" onFocus="this. Class-
    Name='input2'" value="搜产品" onClick="this.value='';"onblur="this.
    value = this.value =='' ? '资源名称' : this.value;this.className='input'" />
</div>
<input class="but" type="submit" value=" " /></form>
</div>
<div class="other" style="left: 200px">
<?php
    $g->loginStats ();
?>
</div>
</div>
</div>
</DIV>
<DIV class=wrap>
<DIV class=left>
<DIV class=commend>
<DIV class=group>
<DIV class=title>产品列表</DIV>
<DIV class=user>
<?php
//表单处理代码
if(isset($ POST["do"])){
    $action = strval($_POST["do"]);
    $error = "";
    switch($action){                                    //判断用户提交的约定变量
        case "over":
            $total = count($_POST["cid"]);
            if($total>0){
                foreach($ POST["cid"] as $k=>$v){   //遍历用户提交的表单变量
                    $g->setSql("update # order set 'over' = 1 where id =
                    '".$v."'");
                    $g->query();                    //运行定单状态更新的 SQL 语句
                }
                echo '共发货'.$total.'件，发货成功！';
            }else{
                echo '请选择货品记录';
            }
        break;
    }
}
?>
<!---------列表开始--------->
<?php
```

```
$header = array("ID","产品名称","市场价","商城价","推荐");//设置通用表格类表头
$g->setSql("select b.id,a.title,a.price,a.mprice,a.commend from # product
a,# order b where a.id = b.pid and b.over = 0");
$g->query();
$data = $g->loadRowList(1);                          //获取定单数据
$toolbar = array(
    "over"=>array("value"=>"完成发货","action"=>"")
);
//设置通用表格类工具类
echo $table->normal($header,$data,true,$toolbar);  //显示定单列表
?>  <!---------列表结束---------></DIV>
</DIV>
<DIV class=space>
<DIV class=title>说明</DIV>
<form name="newFolder" id="newFolder" method="post"
    action="user product.php" onSubmit="return checkSubmit(this);">
<UL class=cool>
    <br>
    <div>选择需要发货的记录，单击完成发货按钮，完成发货动作。</div>
</UL>
</form>
</DIV>
</DIV>
</DIV>
<DIV class=right>
<DIV class=play>
<DIV class=title>登录用户信息</DIV>
<DIV class=playwrap>
<UL id=scrollPlay>
<?php
$g->UserInfo ();
?>
</UL>
</DIV>  </DIV>  </DIV>  </DIV><DIV class=clear></DIV>  </DIV>
</DIV>
```

单击“结算”按钮，或者在浏览器中执行 http://localhost/b2c/user_order.php，运行后界面如图 21.22 所示。



图 21.22　订单管理页面

【代码解读】

user_order.php 的代码主要包括以下两个部分。

（1）订单数据处理代码：当检测到本页有 POST 提交数据时，对产品进行件数统计，并修改订单数据表 over 字段的值，来修改发货状态。

（2）订单数据列表：使用通用列表类，通过组合 SQL 语句，列出没有经过处理的订单。

当选择订单列表中的记录，单击“完成发货”按钮后，系统将统计已经发货的件数，并再次跳转到订单管理页面。这时已经发货的产品，就不会出现在列表中了。

在订单管理页面勾选产品选项，然后单击“完成发货”按钮，完成发货操作。完成发货界面预览效果如图 21.23 所示。



图 21.23　完成发货界面

21.8　产 品 搜 索

在众多产品中，快速地查找到想要的产品，可以节省会员大量的时间。本节将根据这个功能，实现产品搜索页面。下面具体介绍产品搜索的设计和实现步骤。

（1）技术要点：在构建页面时，每一页都创建了一个搜索表单，用于向 search.php 程序文件以 POST 方式提供一个变量。在 search.php 中，根据这个变量，创建 SQL 语句，以查询与变量相关的产品。

（2）产品搜索的实例代码：首先来看搜索表单的设计，该表单用来提供用户输入关键字并传送给 Search.php 程序处理搜索结果。具体表单代码如下：

```
<form action="search.php" method="post" name="search" target=" blank">
<div class="input">
    <input type="hidden" name="type" value="file" />
    <input class="input" name="keyword" type="text"
    onFocus="this.className='input2'" value="搜产品" onClick="this.
    value='';"
    onblur="this.value = this.value ==' ' ? '资源名称' : this.value;this.
    className='input'" />
</div>
<input class="but" name="searchButton" type="submit" value=" " />
</form>
```

搜索页面程序保存在根目录下，文件名为 search.php。详细代码如下所示。

```
<?php
session_start (); //初始化 SESSION 数据
include 'global.php'; //引用公用文件
?>
........//参考随书光盘
<?php
$where = "";
if (isset ( $_POST ["keyword"] )) {     //获取用户提交的约定变量
    $keyword = strval ( $_POST ["keyword"] );
    $where .= " and (title like '%" . $keyword . "%' and intro like '%" .
    $keyword . "%')";
}
$g->setSql ( "select id,title,price,mprice,commend from # product where 1
" . $where . " order by id desc" );     //设置查询使用 SQL 语句
$g->query ();
if ($g->getLines () > 0) {              //当查询结果数大于 0 时，显示结果
    //设置表格类使用的参数
    $header = array ("编号", "名称", "市场价", "商城价", "推荐" );
                                        //设置通用列表类表头
    $data = $g->loadRowList ( 1 );      //获取产品数据
    $toolbar = array ("cart" => array ("value" => "放入购物车", "action" =>
    "cart.php" ), "detail" => array ("value" => "查看详情", "action" =>
    "store.php" ) );                    //设置通用列表类工具栏
    $table->_name = "productTable";//设置通用列表类表单名称，防止与其他表单冲突
    echo $table->normal ( $header, $data, true, $toolbar );
                                        //显示查询结果
} else {
    echo '<LI>暂无产品</LI>';
}
?>
........//参考随书光盘
```

运行搜索程序（search.php）后，执行结果界面如图 21.24 所示。



图 21.24　搜索结果页面

【代码解读】

接收表单提交的变量，并根据变量组成包括 SQL 语句的字符串。使用通用列表类，列出 SQL 语句查询出来的结果，形成类似产品列表的页面。

21.9　小　　结

本章介绍的在线购物平台模块属于电子商务应用领域的 B2C（Business To Customer）模式的典型应用。本章从在线购物平台的系统概述讲起，第 21.1 节介绍在线购物的流程和需要实现的功能，并做了必要的说明。然后根据前面的功能规划指定了文件目录结构设计及相应的说明。第 21.2 节介绍数据部分的设计，详细介绍了本章用的 4 张表，分别是产品数据表、产品附件数据表、会员数据表、订单数据表。第 21.3 节对在线购物平台所用到的核心程序做了详细说明。由于后面的业务程序的功能都是在此之上搭建的，因此读者应重点理解这部分内容，主要包括通用表格类、购物车类、文件上传类，图片显示相关类。第 21.4 节介绍了在线购物平台的业务逻辑部分会员管理。会员管理分为 3 个部分，即会员注册、会员登录和会员资料管理界面。第 21.5 节介绍了产品信息管理和产品图片管理。第 21.6 节介绍了产品展示与购物车。本节从购物车讲起，然后分别介绍了产品详细页实现和产品列表。第 21.7 节和第 21.8 节介绍了在线购物平台的两个辅助功能，即订单的管理和产品搜索。以上部分构成了一个完整的在线购物平台的模块。

第 22 章　Joomla 快速建站模块（PHP+Joomla+XML）

Joomla 快速建站模块是 Joomla 开源平台的具体应用。本章首先介绍 Joomla 的基础知识，然后通过讲解 Joomla 的目录结构为读者梳理 Joomla 的架构。然后开始下载和安装程序，通过实际的操作为读者介绍如何设置管理系统、如何向系统中添加内容。在熟悉了基本的平台功能后开始进入高级的开发阶段：对系统的扩展开发，最后通过一个快速建站的实例，将前面的内容作完整的回顾。

通过本章的学习读者可以了解到 Joomla 的基础知识、平台的基本操作，以及如何开发 Joomla 的相关扩展功能，并具备使用 Joomla 平台快速搭建网站的能力。

本章主要涉及的知识点如下所示。

- Joomla 是什么：Joomla 是目前被广泛使用的、免费的、开源内容管理系统（CMS）。
- “三层架构（3-tier application）”的设计模式：通常意义上的三层架构就是将整个业务应用划分为表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL）。
- Joomla 中的角色划分：前端用户分为注册用户组、作者角色（Author）、编辑角色（Editor）、发布者角色（Publisher）；后台用户分为管理员角色、高级管理员角色、超级管理员角色。
- Joomla 的 3 类扩展：组件（component）、模块（module）、插件（plugin）。

22.1　Joomla 基础

Joomla 是目前被广泛使用的、免费的开源内容管理系统（CMS）。Joomla 的主要特色功能包含可提高效能的页面高速缓存、RSS 馈送、页面的可打印版本、新闻摘要、博客、投票、网站搜索与语言国际化。由于 Joomla 模块化管理和便于扩展的特性，使得它成为中小企业建站的理想选择。

22.1.1　Joomla 介绍

Joomla 是使用 PHP 语言加上 MySQL 数据库所开发的内容管理系统，它可以在 Linux、Windows、Mac OS X 等各种不同的平台上执行。

1．生产环境配置要求

目前，Joomla 最新的版本是 Joomla1.6，生产环境的 PHP 版本至少要达到 5.2.4 以上、

MySQL 需要 5.0.4 以上。Joomla 的开源标志 Logo 如图 22.1 所示。



图 22.1　Joomla 标志 Logo

注意：新版本的 Joomla 不再有 Legacy 模式。

2. Joomla的扩展特性

Joomla 能够称为一款优秀的内容管理软件，除了上面提到的诸多特色以外，另一个重要原因就在于它支持第三方扩展（Extensions）。通常的情况下有了这些扩展可以快速的搭建常见的应用，如论坛、留言本、wiki、Blog（博客）等，在 Joomla! 官方扩展库（extensions.joomla.org）里面，可以找到各种需要的扩展程序，这些程序极大的丰富了 Joomla 的应用也为系统扩展提供更大的灵活性。

Joomla 的扩展分 3 类：

（1）组件（component）：实现某个重要的、复杂功能的程序集。组件展示的位置是在模板的“mainbody”区域。组件可以在前台菜单上建立菜单项，单击该菜单项就打开了该组件的内容。

（2）模块（module）：模块无法独立成页，它必须被安排在页面上非“mainbody”区域。在一个模块位置上，可以发布多个模块，可以在后台调整展示顺序。

（3）插件（plugin）：一种即时运行的程序，由触发代码启动，效果作为“网页、网站”的一部分来展示。它们一般有自己的使用语法，大多是用大括号“{}”括起来的一段符合独有的语法格式的标记。

注意：在 1.5 版本以后，触发器被称为触发器（mambot）。

除了官方提供的各种扩展库以外，还可以根据规则创建自己的 Joomla 扩展。关于创建组件（component）、模块（module）和插件（plugin）扩展，会在本章的后面结合具体的例子为读者做详尽的介绍。

22.1.2　Joomla 整体架构

首先介绍一个软件工程学上的概念，即“三层架构（3-tier application）”的设计模式。通常意义上的三层架构就是将整个业务应用划分为表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL）。这种划分的目的是为了实现软件工程学上的“高内聚，低耦合”的设计原则。

同样，Joomla 在代码组织上也采用 3 层架构的模式。最底层是框架层，包含一些核心

的库及一些插件；第二层是应用层，包括 JAppication 类，目前随 Joomla 发布的有 3 个类 JInstallation、JAdministrator 和 JSite，应用层是页面的主要控制器；第三层是扩展层，在这一层包含所有的组件，模块和模板被执行和渲染（render）。Joomla 的 3 层架构如图 22.2 所示。

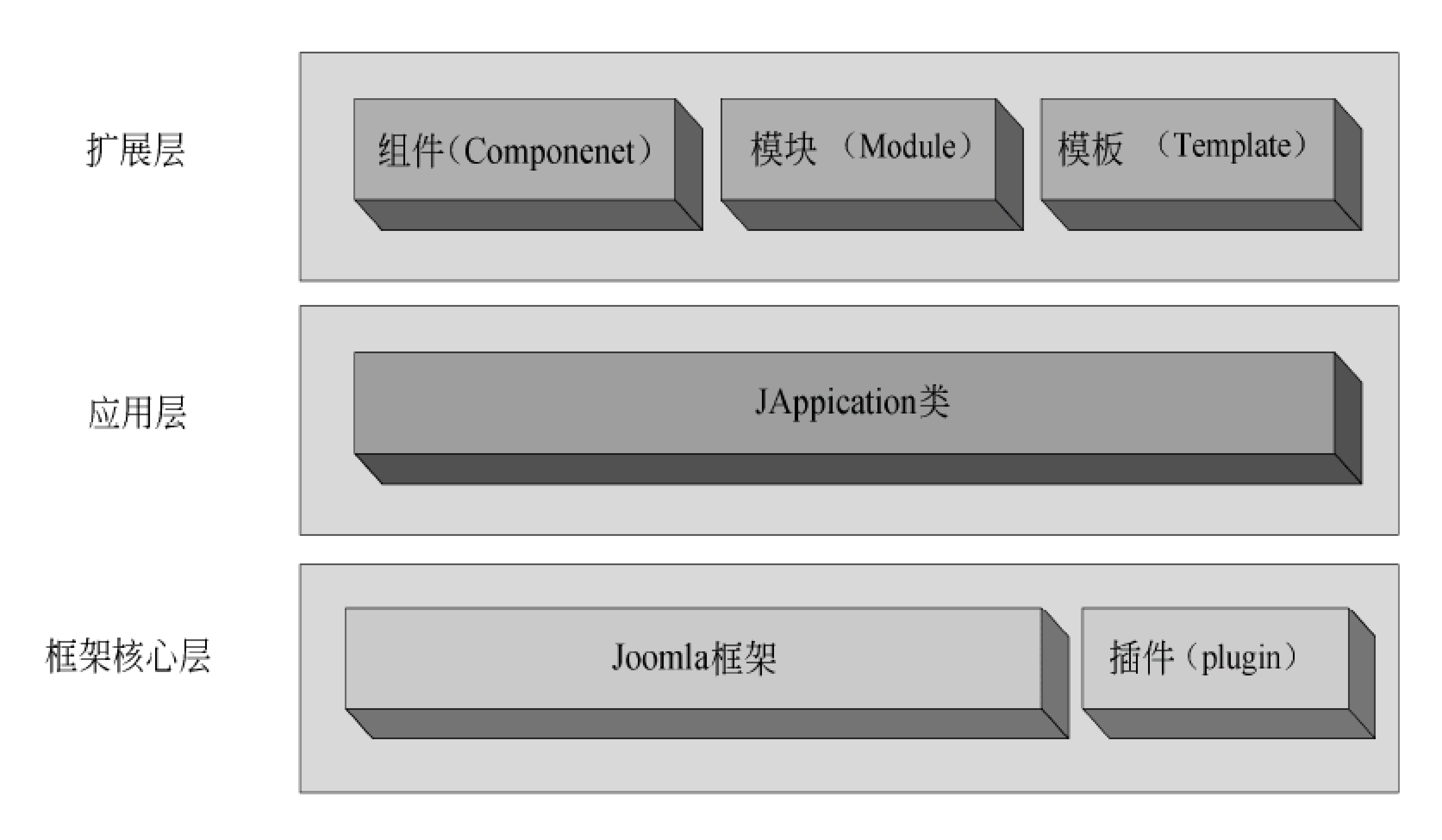


图 22.2　Joomla 的 3 层架构图

下面是 Joomla 官方网站列出的框架核心层中涉及的重要包及其含义。

- Application Package：JApplication 相关库。
- Cache Package：缓存控制。
- Common Packag：兼容，以及继承等一些基本类。
- Connector Packag：连接器，包括 FTP、LDAP 等。
- Database Package：数据库相关类。
- Document Packag：构造和渲染页面的相关函数，其实就是指 HTML 页面。
- Filesystem Package：与文件系统交互的类。
- i18n Package：国际化类库。
- Installer Package：安装类库。
- Mail Package：邮件相关类。
- Model Package：数据访问类。需要注意的是，与 Jdatabase 类不同（JDatabase 负责数据库链接和预计执行），Model 是数据视图。
- Parameter Package：参数存取相关类。
- Registry Package：配置文件处理相关类。
- Template Package：模板类。
- Utilities Package：工具类。

以下是核心类及其含义：

- JFactory 类：工厂类，生成各种重要的对象。
- JVersion 类：版本控制类。

由于应用层和扩展层涉及的内容众多，请读者参考官方网站提供的相关文档，这里不再赘述。

22.2　Joomla 目录及文件说明

前文介绍了 Joomla 的基础知识并简要地介绍了 Joomla 整体架构，本节将说明 Joomla 目录和文件的具体含义。通过本节的介绍便于读者了解 Joomla 的组成，为后面介绍的插件、组件和模块开发打下基础。

22.2.1　根目录说明

在官方网站下载 Joomla 压缩包，解压到本地，打开文件根目录如图 22.3 所示。下面介绍根目录中的文件夹及文件的具体含义。

根目录下的文件夹含义如下。

- /administrator：后台管理目录，用来存放后台管理相关文件。
- /cache：缓存目录，用来存储程序运行过程中产生的缓存文件，缓存的作用是能够提高网站的访问速度。
- /components：组件目录。
- /images：存放图片的目录。
- /includes：公用文件目录。该目录中存放运行所需要的各种脚本、函数等。
- /language：网站前台语言包目录。后台语言包放置于/administrator/language 目录下。
- /libraries：类目录。该目录中存放着重要的类文件，子目录结构对应类名称。
- /logs：日志目录。存放生成的日志脚本文件。
- /media：媒体文件目录。存放各种媒体文件（用来给网站的用户下载）。
- /modules：模块目录。存放模块程序，用来将特定的内容展示在前台“mainbody”之外的指定位置的程序。
- /plugins：插件目录。每个插件都是独立运行的程序，由代码来触发。效果由网站的一部分来展示。
- /templates：模板目录。里面可以建立多个子目录，每个子目录对应一种模板。
- /xmlrpc：xml 远程方法调用。采用 XML 作为调用远程方法的媒介。
- /tmp：临时文件目录。

下面介绍根目录下几个重要的文件的含义。

- configuration.php：Joomla 实例化配置文件。该文件包含程序安装的基本信息，如网站名称、配置路径等。由于该文件中包含数据库的链接信息，因此需要注意文件权限设置。
- Index.php：该文件是程序的入口文件。
- INSTALL.php：初始安装引导文件。

22.2.2　模板目录说明

在前面介绍了 Joomla 的根目录的文件及文件夹含义，下面来介绍核心的、常用的几个

文件夹的功能。Joomla 是按照 MVC 模式组织代码的，因此页面模板代码是以单独文件包的形式安装的，安装后的模板文件存放在根目录的 templates 文件夹下。在新版本的 Joomla 中，允许设计者为模板定制可选参数，这些参数对应不同的选项，例如修改模板背景颜色、字号大小、宽度等，用户可以在管理后台的模板管理中修改相关选项。

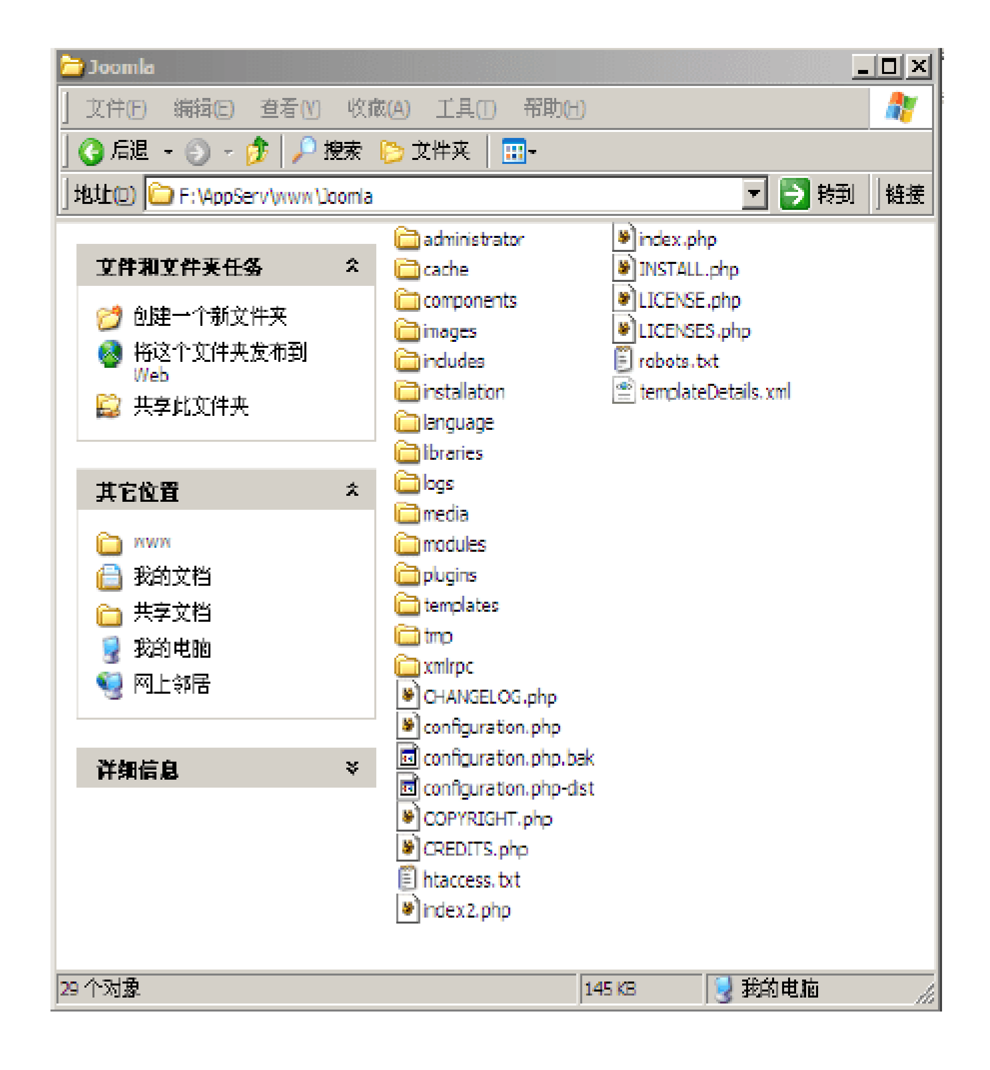


图 22.3　Joomla 根目录

开发者可以定制不同的模板。打开 Joomla 根目录下的 templates 文件夹，该文件夹在默认的情况下包含 3 个前端页面模板文件夹（beez、ja_purity、rhuk_milkyway）和一个系统模板文件夹（system），如图 22.4 所示。

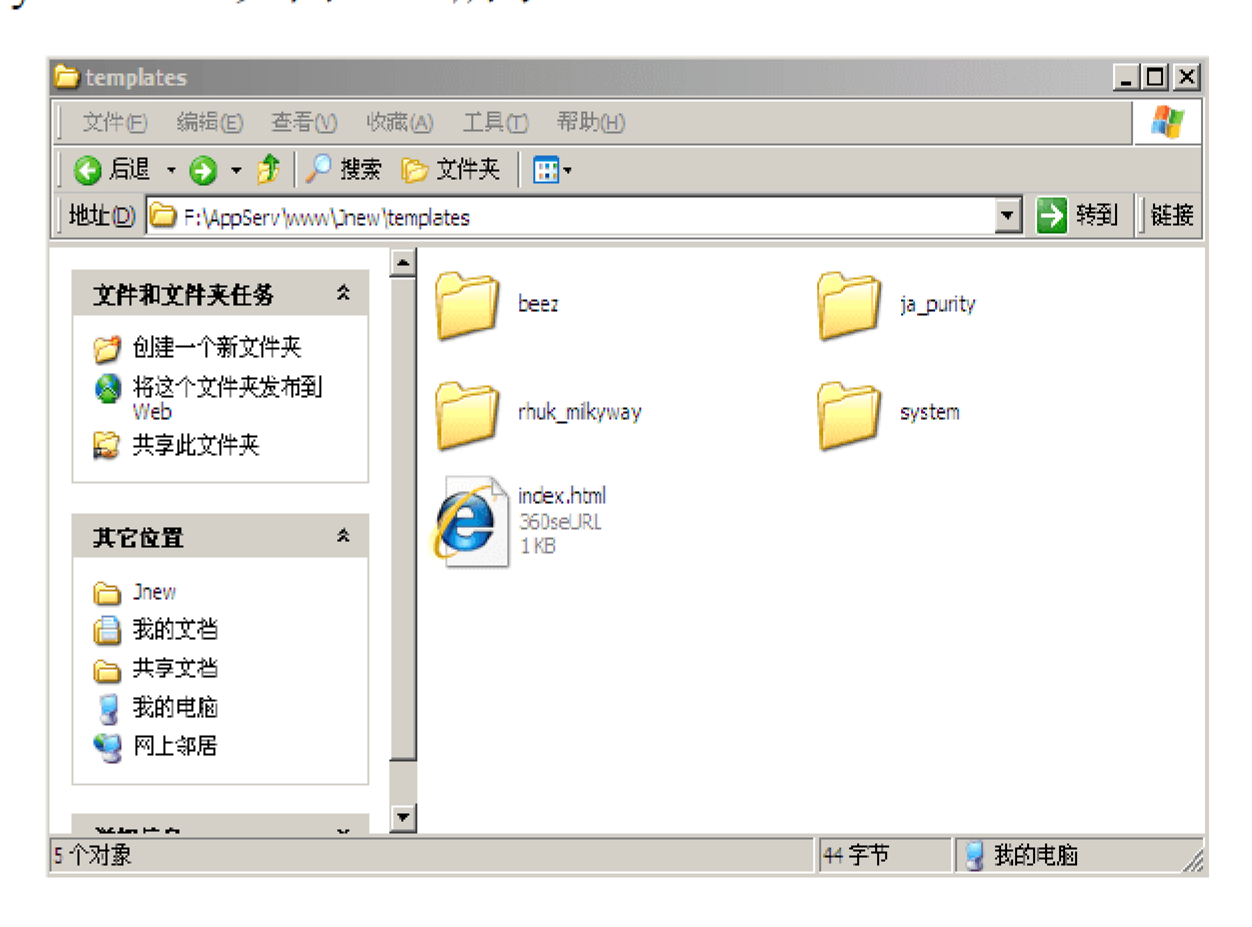


图 22.4　templates 模板文件夹

下面以其中的一个页面模板文件夹（ja_purity）为例，介绍模板文件夹所包含的内容及对应文件的功能。打开 ja_purity 文件夹，如图 22.5 所示。

下面介绍页面模板文件夹（ja_purity）的具体含义。由于 Joomla 中规定页面模板文件都有标准的格式，因此对该页面模板文件夹的解释具有普遍的意义。读者可以参照该文件夹的含义，了解其他页面模板文件的构成。



图 22.5　ja_purity 模板文件

在典型模板目录结构中，模板文件夹中至少包含以下 4 个文件。

- index.php 文件：提供模块和组件的显示。
- template.css 文件：处理模板的显示方面，包括边缘、字体、图像边框、列表格式等。
- templateDetails.xml 文件：模板文件的信息源文件，包含模板安装和管理过程中用到的配置信息。
- template_thumbnail.png 文件：风格模板缩略图（在模板管理中，当光标停留在模板名字上时，通常会显示一个 200×150 像素的图像。这给站点管理员应用模板前提供一个快照视图）。

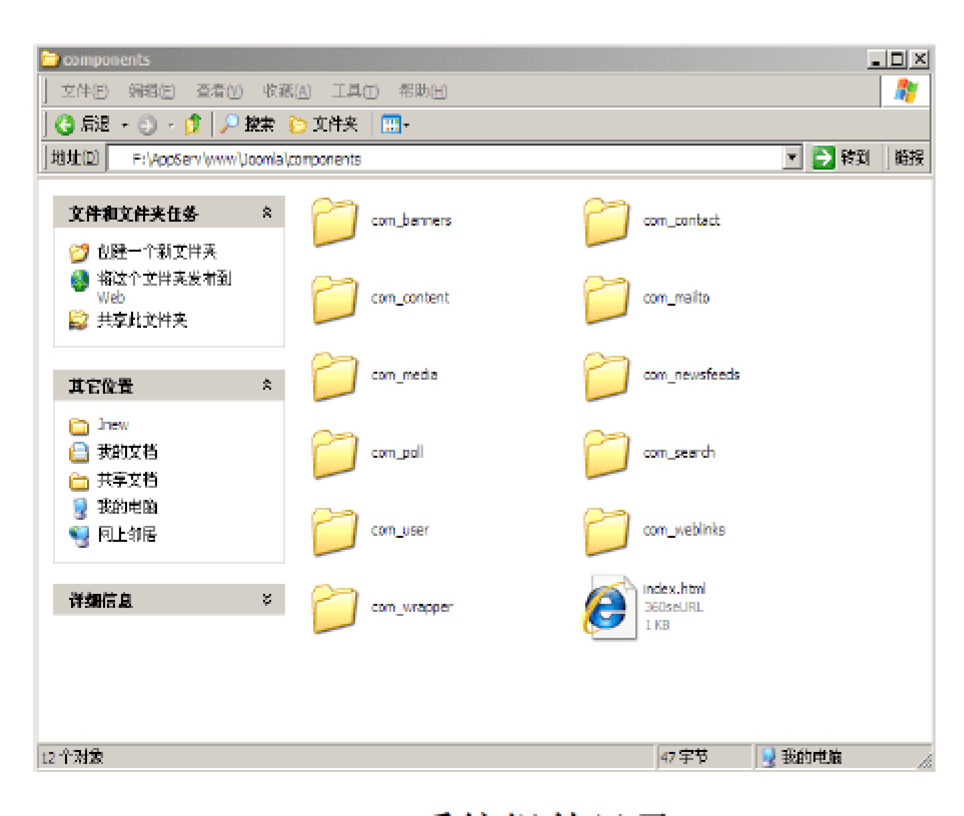
典型的目录还包括以下文件夹。

- Css 文件夹：包含所有的 css 文件。
- Html 文件夹：包含核心输出与模板覆写文件。
- Images 文件夹：包含模板使用的所有图像。

22.2.3　组件目录说明

在 Joomla 的 3 类扩展中，“组件”是最为重要、最为强大的扩展。它可以生成自己的前台页面（也就是访问者能看到的网页），也可以拥有自己的后台配置页面（供网站管理员使用）。因此，可以把组件看成被整合到 Joomla 平台中的独立的应用程序。

除了 Joomla 的核心管理和发布功能以外，很多重要功能都是通过“组件”扩展的，如博客、论坛、图库、社区、在线商城、OA、CRM 等各类应用系统。系统默认组件文件如图 22.6 所示。



22.6　系统组件目录

下面介绍系统默认的组件的用途及功能，详细信息如表 22.1 所示。

表 22.1　系统默认组件介绍

组 件 名 称	用途及功能
com_banners	广告内容管理组件
com_content	内容分类管理组件
com_media	媒体管理组件
com_poll	投票调查管理组件
com_user	用户管理组件
com_wrapper	嵌入模式运行页面组件
com_contact	“联系我们”管理组件
com_mailto	邮件管理组件
com_newsfeeds	新闻联播管理组件
com_search	用户搜索组件
com_weblinks	友情链接组件

前面介绍了 Joomla 系统默认组件的用途和功能，下面以“友情链接”组件为例，介绍一个基本的组件的构成。打开组件目录下的 com_weblinks 目录，如图 22.7 所示。

“友情链接”组件（com_weblinks）文件夹下文件及目录的含义如下。

- Controller 目录：控制器文件目录，该目录中包含了控制器文件类（weblinks.php）。
- Models 目录：模型目录，该目录中每一个模型类对应数据库中的一个数据表。
- Views 目录：视图目录，该目录中包含视图类和模板文件（一个视图可能对应多个模板文件，分别保存在 tmpl 目录中）。
- metadata.xml：组件加载配置文件。
- controller.php：默认的控制器入口文件。
- weblinks.php：组件入口文件。

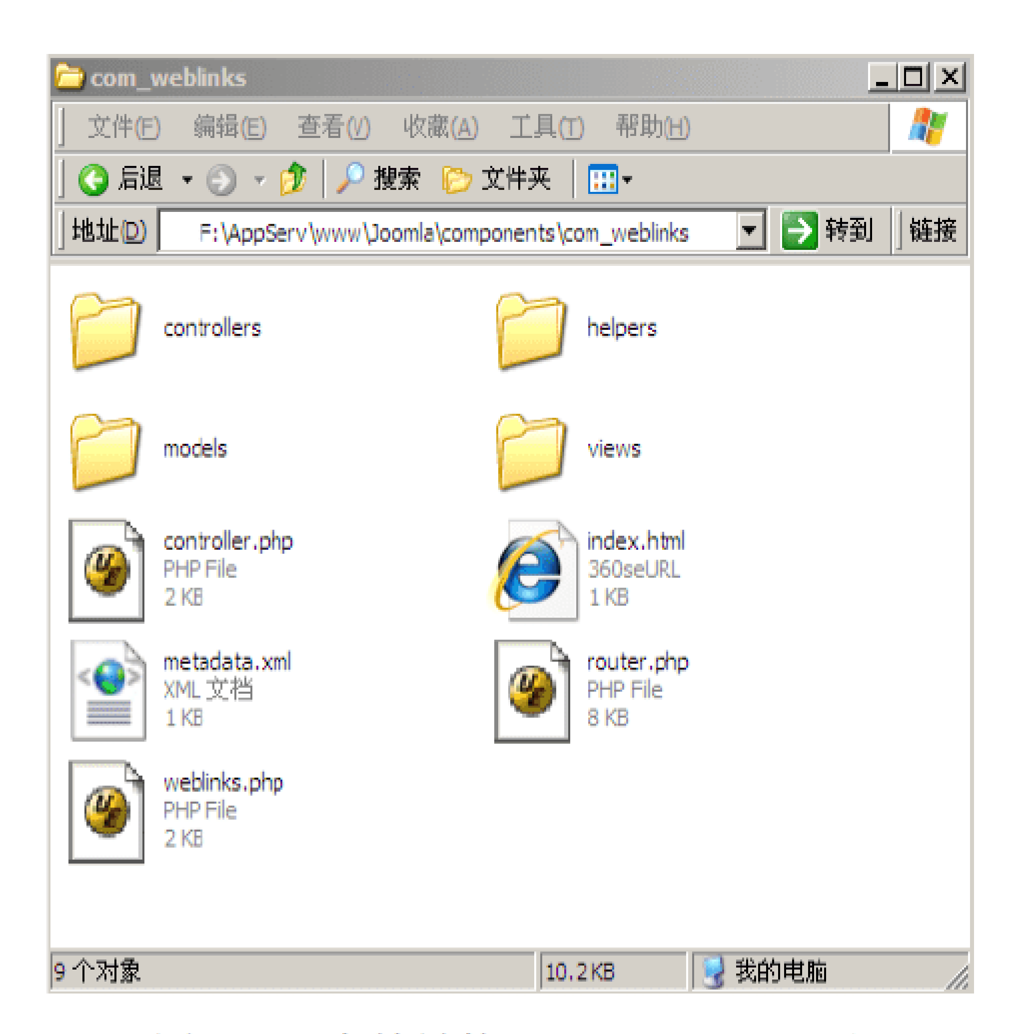


图 22.7　友情链接 com_weblinks 目录

22.2.4　模块目录说明

在前文介绍过 Joomla 中的模块是一种可视化的扩展。在页面上呈现的内容都可以被看

作是模块，如搜索框、登录框、导航栏、Banner 幻灯片、热门文章、最新文章、RSS 新闻订阅等。

由于模块无法独立成页，它必须被安排在页面上非“mainbody”区域。在一个模块位置上，可以发布多个模块，可以在后台调整展示顺序。模块目录如图 22.8 所示。

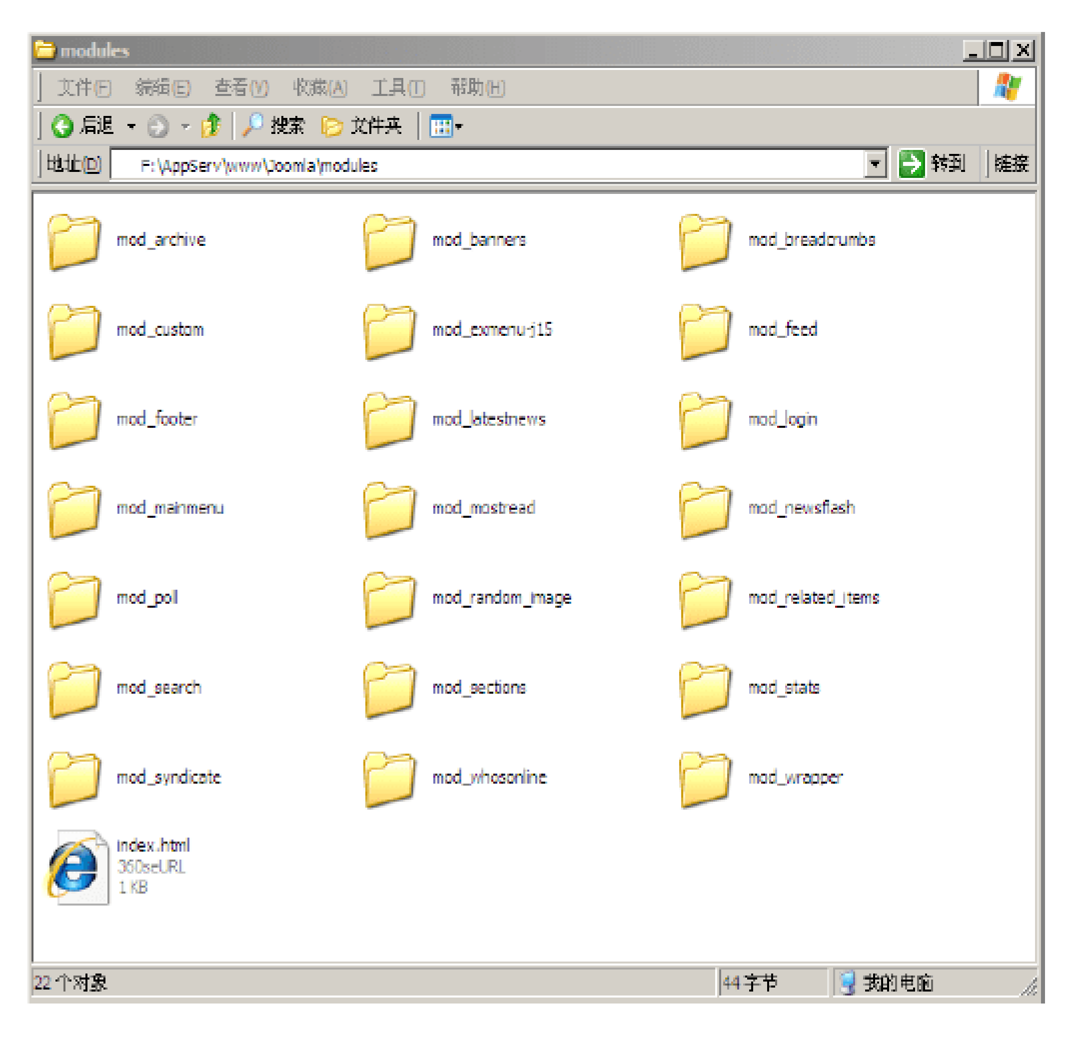


图 22.8　系统默认模块目录

下面以投票调查模块（mod_poll）为例，介绍 Joomla 中模块目录中的文件组成及功能含义。mod_poll 模块目录文件内容如图 22.9 所示。

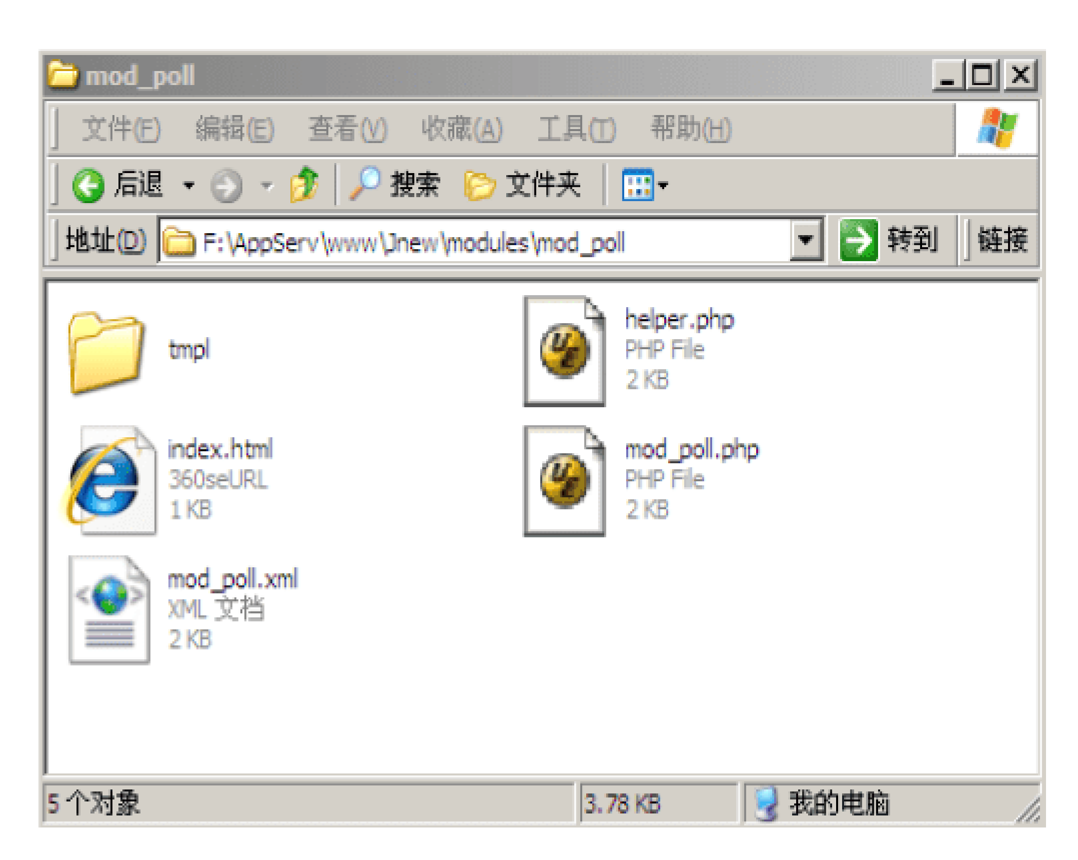


图 22.9　投票调查模块

mod_poll 模块目录文件内容含义如下。

- tmpl 目录：模块对应的模板文件。
- mod_poll.php：模板的主文件，通过该文件来调用模板。
- helper.php：包含模块帮助类。
- mod_poll.xml：模块需要加载的配置文件（XML 格式）。

22.2.5　插件目录说明

根据前面的介绍，Joomla 中的插件（plugin）是一种即时运行的程序，由触发代码启

动，效果作为“网页、网站”的一部分来展示。它们一般有自己的使用语法，大多是用大括号“{}”括起来的一段符合独有的语法格式的标记，由于插件的开发方式十分灵活，有时候就是一个程序段，因此这里不做详细的说明。插件目录如图 22.10 所示。

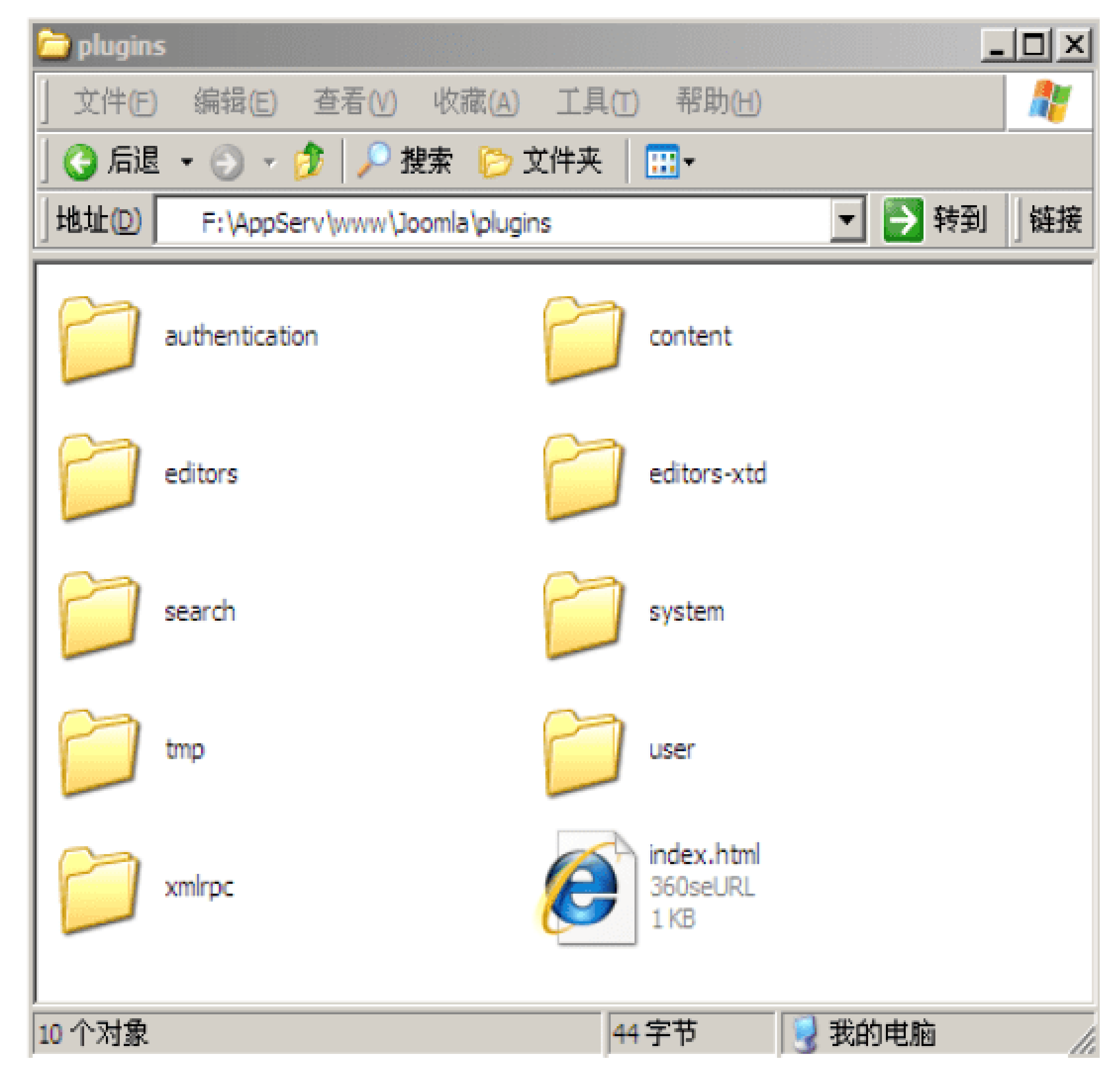


图 22.10　Joomla 插件目录

22.3　Joomla 数据库

通过前文的介绍了解到，Joomla 的典型应用模式是通过 Web 的方式对数据库进行交互。因此了解任何关于 Joomla 的应用，首先都应该从认识 Joomla 的核心数据库开始。本节将为读者介绍 Joomla 数据库的基础知识，以及各个表之间的逻辑关系。

22.3.1　Joomla 数据库基础

Joomla 数据库安装成功之后，创建基础的表一共 36 个左右（根据版本不同会略有差异），如图 22.11 所示。这些基础表大致可以分为两类。

- 第一类是核心框架表（如 jos_content 表是核心的内容管理组件的数据表）；
- 第二类是核心框架的扩展（如使用到 phpacl 权限框架表 jos_core_acl_aro）。

1．数据库初始操作

在 Joomla 中规定，获取 Joomla 数据库是通过全局对象 JDatabase 类来实现的。JDatabase 是一个抽象类，各种数据库驱动都是继承于这个类（如 Oracle、SQL Server 等），读者可以参考 libiraries/joomla/database/database/mysql.php 来编写对应的数据库驱动。通过数据库操作类（connector）来操作数据库，该类中封装了与数据库操作相关的基本操作。具体可以通过两个子类的操作来实现，即 JDatabaseMysql 类和 JDatabaseMysqli 类。connector 类为以后扩展到其他数据库留出了接口，采用这种能够使系统使用几种不同的数据库系统。

JDatabase 类自动创建数据库链接，仅要简短的代码就获得返回结果，十分高效。同时，采用这种数据层的设计，最大限度地确保了系统的灵活性和兼容性。

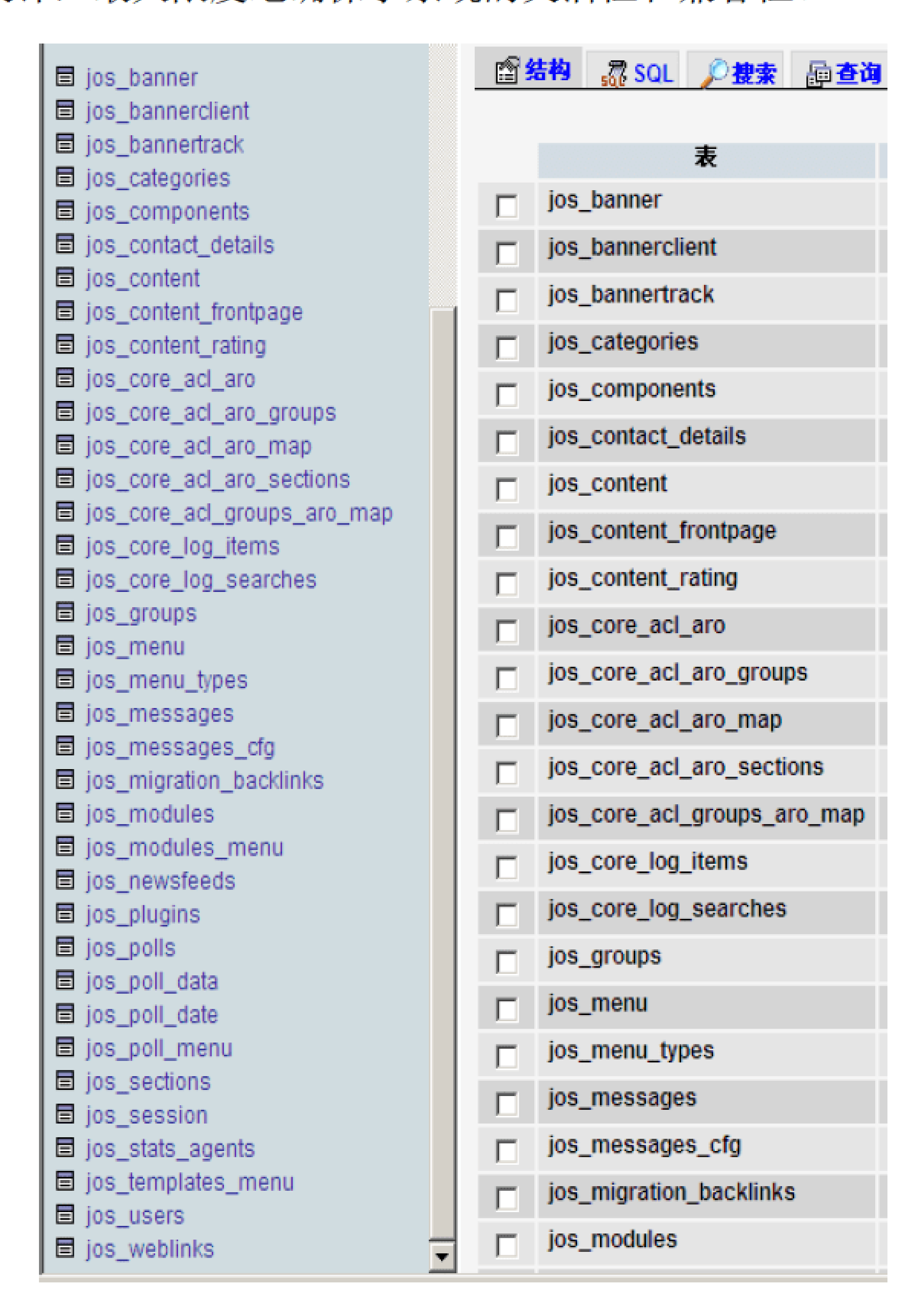


图 22.11　Joomla 基础表

下面通过一个数据库查询的例子，为读者直观地说明数据层的执行流程。代码部分如下：

```
$db = JFactory::getDBO();// 获取数据库对象
$query = "SELECT * FROM #__example_table WHERE id =XXX";
$db->setQuery($query);
```

【代码解读】

首先初始化数据库对象，然后设置查询语句。可以使用通常的 SQL 语法，唯一需要改变的是表前缀。然后执行 setQuery()方法。

注意：为了使尽可能的灵活，Joomla 为前缀采用了一个占位符#__。

标准流程的执行步骤如下。

（1）设置 setquery()方法（配置查询语句等）；

（2）加载 load 或者直接执行 executequery()方法（加载或者执行查询语句）。

这里需要注意的是，以 load 开头的几个方法的具体功能的区分。

- loadObject()方法：以对象的形式返回结果集的第一行数据。
- loadObjectList()方法：对应上一个函数，返回一个对象的集合。

- loadResult()方法：返回第一行的第一个字段或者 NULL。
- loadResultArray()方法：返回某一字段的数据到一个数组中。
- loadRow()方法：返回第一行数据，应该是数组形式。
- loadRowList()方法：对应上一个函数，返回行数据的集合。

注意：还有一个很重要的 setutf()方法，该方法的作用是设置数据库的默认字符集。

获取全局对象 JDatabase 的代码如下：

```
$db = JFactory::getDBO();//获取数据库对象
```

2．表前缀

良好的数据库设计习惯之一是在数据库表中使用统一的前缀，Joomla 默认安装情况下的前缀名是 jos_。当数据库系统中只有一个库的时候前缀没有太多的意义，但是当有多个项目使用同一个公用库的时候表前缀就会变得很重要。一般单个的 Joomla 表使用系统默认的 jos_前缀来命名（Joomla 规定，在程序中设置 SQL 语句时，使用#__来代替前缀名）。

注意：当安装时设置了数据库中表的前缀名后，可以通过修改“全局设置”来修改数据库前缀命名。当出现异常时，可以修改 configuration.php 文件中第 37 行（不同版本略有不同），修改为原来的代码：

```
var $dbprefix = 'jos_';
```

3．数据库结构约定

Joomla 数据库的标准中，除了规定通常需要遵守的命名规范外，还约定其他一些需要遵守的规则。由于这些约定的规范涉及 Joomla 框架结构中的一些自动操作和查找功能，如果不遵守规则。这部分功能就无法实现。

首先所有的表必须使用表前缀，然后是该插件的扩展名。如果这个表是用于存储一些实体数据的，还需要加上实体名来与该插件下的其他实体数据表相互区别。3 者之间用下划线来连接，比如说在扩展 MyExtension 中存储 items 的表，就可以写为 #__MyExtension_items。

下面具体介绍数据库中常用字段的规定。

- Publishing 字段（发布）：该字段被约定是否发布或者显示该条数据。该字段使用 published（tinyint(1)）表示，其中 1 表示发布、0 表示不发布。
- Hits 字段（点击）：该字段被约定用来追踪某一条数据被访问的次数。可以使用 hits（Integer 整型）来表示，默认是 0 Checking Out（锁定）。
- Ordering 字段（排序）：该字段被约定提供给管理员自由选择内容项显示顺序的功能。Ordering（Integer 整型）是标记记录的顺序，从而决定是按照何种顺序来显示这些记录，字段的值通常是连接在 where 查询子句后面，用于实现排序规则的。
- 日期字段：该字段被约定使用 Datetime 字段来记录动作发生的日期和时间。当使用该字段的时候需要注意时区的影响，所有日期和时间数据必须被标记为

"UTC+0"格式。当需要使用时，可以调用 Joomla 的 JDate 类来格式化输出该数据。

- 参数字段：该字段被约定使用 text 文本字段描述该记录的一些额外信息。

22.3.2　Joomla 数据库逻辑关系

Joomla 数据库按照逻辑组成可以分成以下 7 个部分，即 content（内容部分）、menu（菜单部分）、template（模板部分）、components（组件部分）、"compents，modules&plugins"（扩展部分）、"user&access control"（用户及权限控制部分），"logs&stats"（日志部分）。这 7 个部分构成了 Joomla 数据库的框架基础，具体的逻辑关系及 Joomla 数据库的组成结构如图 22.12 所示。

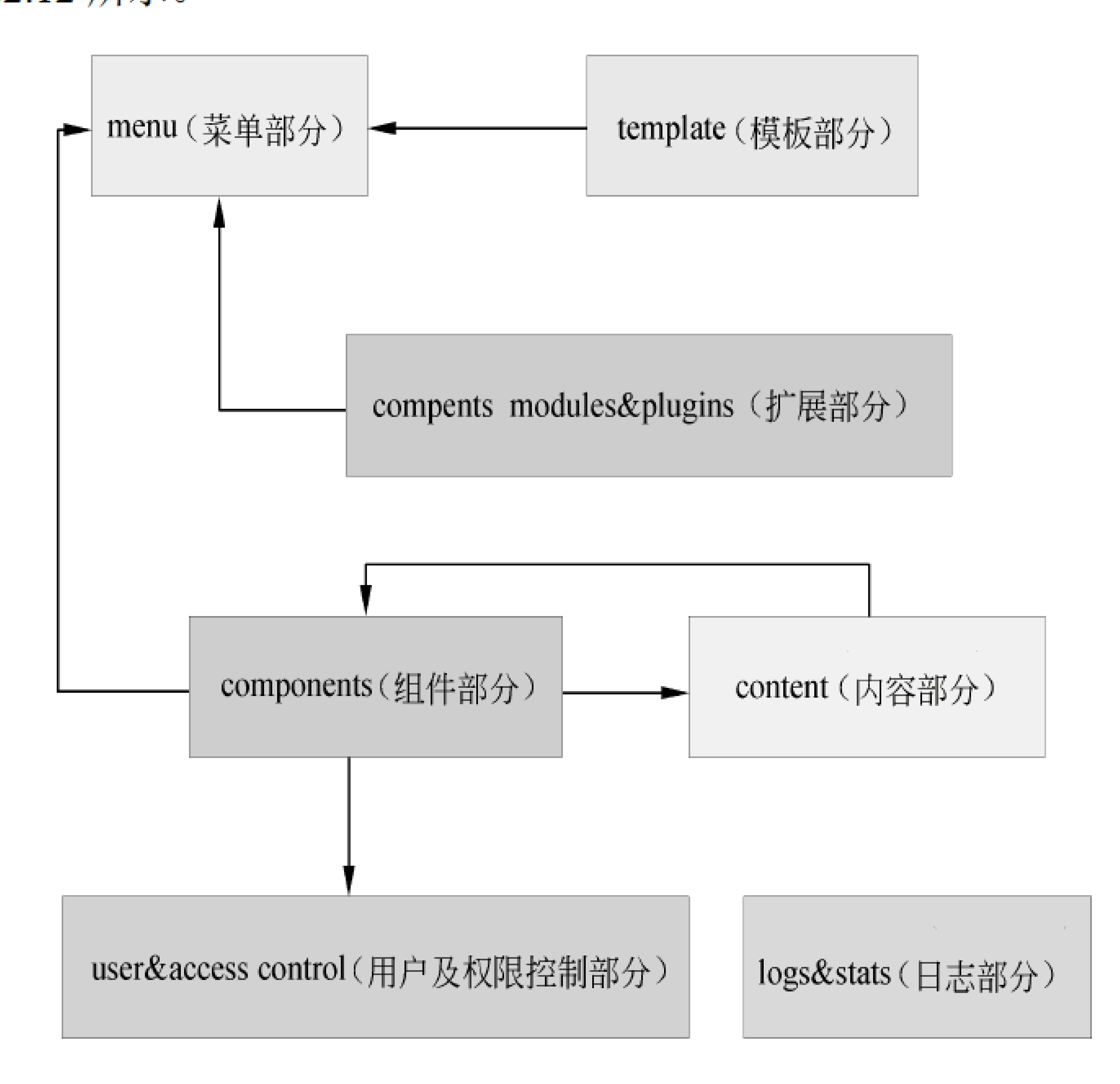


图 22.12　Joomla 数据库组成结构图

22.3.3　Joomla 数据库表说明

Joomla 数据库安装成功之后，创建基础的表一共 36 个左右。由于系统表格众多，因此通过功能分类的方式介绍其中重要的数据表。下面以表格的形式，具体介绍表的数据结构和字段含义。

（1）广告内容管理组件

说明： 广告内容管理组件（com_bannar）对应 3 个表，分别是广告条表（jos_bannar）、广告客户表（jos_bannarclient）和广告追踪表（jos_bannertracks）。这 3 个表（表 22.2～表 22.4）共同管理广告内容关键组件的相关业务逻辑。

表 22.2　广告条表（jos_bannar）

字　段	名　称	类　型	整　理	默　认	额　外
bid	广告 id	int(11)	NULL	（NULL）	auto_increment
cid	客户 id	int(11)	NULL	0	
type	类型	varchar(90)	utf8_general_ci	Banner	
name	名称	text	utf8_general_ci		
imptotal	数量	int(11)	NULL	0	
impmade	制作	int(11)	NULL	0	
clicks	点击数	int(11)	NULL	0	
imageurl	图片地址	varchar(100)	utf8_general_ci		
clickurl	链接地址	varchar(200)	utf8_general_ci		
date	日期	datetime	NULL	（NULL）	
showBanner	是否显示	tinyint(1)	NULL	0	
checked_out	检验	tinyint(1)	NULL	0	
checked_out_time	检验时间	datetime	NULL	0000-00-00 00:00:00	
editor	编辑人	varchar(150)	utf8_general_ci	（NULL）	
custombannercode	客户广告条代码	text	utf8_general_ci	（NULL）	
catid	分类 id	int(10) unsigned	NULL	0	
description	描述	text	utf8_general_ci		
sticky	附件	tinyint(1) unsigned	NULL	0	
ordering	排序	int(11)	NULL	0	
publish_up	发布日期	datetime	NULL	0000-00-00 00:00:00	
publish_down	结束日期	datetime	NULL	0000-00-00 00:00:00	
tags	标签	text	utf8_general_ci		
params	参数	text	utf8_general_ci		

表 22.3　广告客户表（jos_bannarclient）

字　段	名　称	类　型	整　理	NULL	键名	默认	额外
cid	客户 id	int(11)	NULL		PRI	（NULL）	auto_increment
name	名称	text	utf8_general_ci				
contact	联络方式	text	utf8_general_ci				
email	邮件地址	varchar(255)	utf8_general_ci				
extrainfo	额外信息	text	utf8_general_ci				
checked_out	检查	tinyint(1)	NULL			0	
checked_out_time	检查时间	time	NULL	YES		（NULL）	
editor	编辑	varchar(150)	utf8_general_ci	YES		（NULL）	

表 22.4　广告追踪表（jos_bannertracks）

字　段	名　称	类　型	整　理	NULL	键　名	默　认	额　外
track_date	追踪日期	date	NULL			0000-00-00	
track_type	类型	int(10) unsigned	NULL			0	
banner_id	广告条幅 id	int(10) unsigned	NULL			0	

（2）系统分类表

说明：系统分类表（jos_categories），对应存放系统的分类信息。

表 22.5　系统分类表（jos_categories）

字　段	名　称	类　型	整　理	NULL	键名	默　认	额　外
id	Id	int(11)	NULL		PRI	（NULL）	auto_increment
parent_id	父 id	int(11)	NULL			0	
title	标题	text	utf8_general_ci				
name	名称	text	utf8_general_ci				
image	图片	varchar(255)	utf8_general_ci				
section	单元	varchar(150)	utf8_general_ci		MUL		
image_position	图片位置	varchar(90)	utf8_general_ci				
description	描述	text	utf8_general_ci				
published	是否发布	tinyint(1)	NULL			0	
checked_out	检查	int(11) unsigned	NULL		MUL	0	
checked_out_time	检查时间	datetime	NULL			0000-00-00 00:00:00	
editor	编辑	varchar(150)	utf8_general_ci	YES		（NULL）	
ordering	排序	int(11)	NULL			0	
access	权限	tinyint(3) unsigned	NULL		MUL	0	
count	计数	int(11)	NULL			0	
params	参数	text	utf8_general_ci				

（3）CMS 核心功能内容管理

说明：CMS 核心功能内容管理中对应的 3 个表，分别是内容表（jos_content）、首页内容表（jos_content_frontpage）、内容评选表（jos_content_rating）。

表 22.6　内容表（jos_content）

字　段	名　称	类　型	整　理	NULL	键　名	默　认	额　外
id	Id	int(11) unsigned	NULL		PRI	（NULL）	auto_increment
title	名称	text	utf8_general_ci				
title_alias	别名	text	utf8_general_ci				

续表

字　段	名　称	类　型	整　理	NULL	键　名	默　认	额　外
introtext	介绍	mediumt-ext	utf8_general_ci				
fulltext	正文	mediumt-ext	utf8_general_ci				
state	状态	tinyint(3)	NULL		MUL	0	
sectionid	单元 id	int(11) unsigned	NULL		MUL	0	
catid	分类 id	int(11) unsigned	NULL		MUL	0	
created	创建时间	datetime	NULL			0000-00-00 00:00:00	
created_by	创建人	int(11) unsigned	NULL			0	
created_by_alias	创建别名	text	utf8_general_ci				
modified	修改时间	datetime	NULL			0000-00-00 00:00:00	
modified_by	修改人	int(11) unsigned	NULL			0	
checked_out	检查	int(11) unsigned	NULL		MUL	0	
checked_out_time	检查时间	datetime	NULL			0000-00-00 00:00:00	
images	图片	text	utf8_general_ci				
urls	链接	text	utf8_general_ci				
attribs		text	utf8_general_ci				
version	版本	int(11) unsigned	NULL			1	
parentid	父 id	int(11) unsigned	NULL			0	
ordering	排序	int(11)	NULL			0	
metakey	标记关键字	text	utf8_general_ci				
metadesc	标记描述	text	utf8_general_ci				
access	权限	int(11) unsigned	NULL		MUL	0	
hits	隐藏	int(11) unsigned	NULL			0	
metadata	标记数据	text	utf8_general_ci				

表 22.7　首页内容表（jos_content_frontpage）

字　段	名　称	类　型	整　理	NULL	键　名	默　认	额　外
content_id	内容 id	int(11)	NULL		PRI	0	
ordering	排序	int(11)	NULL			0	

表 22.8　内容评选表（jos_content_rating）

字　段	名　称	类　型	整　理	NULL	键　名	默　认
content_id	内容 id	int(11)	NULL		PRI	0
rating_sum	总分	int(11) unsigned	NULL			0
rating_count	打分次数	int(11) unsigned	NULL			0
lastip	最后登录 ip	varchar(150)	utf 8_general_ci			

（4）联系人组件表

说明： 联系人组件表（jos_contact_details），这个表是对应 com_contact 组件，联系人的详细信息都存在该表中。

表 22.9　联系人组件表（jos_contact_details）

字　段	名　称	类　型	整　理	NULL	键名	默　认	额　外
id	Id	int(11)	NULL		PRI	(NULL)	auto_increment
name	名字	text	utf8_general_ci				
con_position	组件位置	text	utf8_general_ci	YES		(NULL)	
address	地址	text	utf8_general_ci	YES		(NULL)	
suburb	区域	text	utf8_general_ci	YES		(NULL)	
state	状态	text	utf8_general_ci	YES		(NULL)	
country	国家	text	utf8_general_ci	YES		(NULL)	
postcode	邮政编码	varchar(255)	utf8_general_ci	YES		(NULL)	
telephone	电话号码	varchar(255)	utf8_general_ci	YES		(NULL)	
fax	传真号码	varchar(255)	utf8_general_ci	YES		(NULL)	
misc	其他	mediumtext	utf8_general_ci	YES		(NULL)	
image	图片	varchar(255)	utf8_general_ci	YES		(NULL)	
email_to	发送邮件地址	varchar(255)	utf8_general_ci	YES		(NULL)	
default_con	默认	tinyint(1) unsigned	NULL			0	
published	是否发布	tinyint(1) unsigned	NULL			0	
checked_out	检查	int(11) unsigned	NULL			0	
checked_out_time	检查时间	datetime	NULL			0000-00-00 00:00:00	
ordering	排序	int(11)	NULL			0	
params	参数	text	utf8_general_ci				

续表

字　段	名　称	类　型	整　理	NULL	键名	默　认	额　外
user_id	用户 id	int(11)	NULL			0	
catid	分类 id	int(11)	NULL			0	
access	权限	tinyint(3) unsigned	NULL			0	
mobile	移动电话	varchar(255)	utf8_general_ci				
webpage	网站页面	varchar(255)	utf8_general_ci				

（5）模块相关的表

说明：模块相关的表，系统所有安装的组件都在这里，它定义了组件的链接、名称（包括后台管理部分）。在安装一个新组件的时候会自动添加一条记录进来。

注意：当手动删除组件的时候，在对应的表中必须删除相应记录。

模块表（jos_modules）主要记录模块的名称，以及是否发布和发布的 position 位置。模块显示的菜单表（jos_modules_menu）记录模块在哪个菜单上显示就通过这张表控制。

注意：模块在 Joomla 里面是可以指定显示在哪个菜单上的。

表 22.10　模块表（jos_modules）

字　段	名　称	类　型	整　理	NULL	键名	默　认	额　外
id	Id	int(11)	NULL		PRI	(NULL)	auto_increment
title	模块名称	text	utf8_general_ci				
content	内容描述	text	utf8_general_ci				
ordering	排序	int(11)	NULL			0	
position	位置	varchar(150)	utf8_general_ci	YES		(NULL)	
checked_out	检查	int(11) unsigned	NULL			0	
checked_out_time	检查时间	datetime	NULL			0000-00-00 00:00:00	
published	是否发布	tinyint(1)	NULL		MUL	0	
module	模型	varchar(150)	utf8_general_ci	YES	MUL	(NULL)	
numnews	新闻数	int(11)	NULL			0	
access	权限	tinyint(3) unsigned	NULL			0	
showtitle	是否显示名称	tinyint(3) unsigned	NULL			1	
params	参数	text	utf8_general_ci				
iscore	分数	tinyint(4)	NULL			0	
client_id	客户端 id	tinyint(4)	NULL			0	
control	控制	text	utf8_general_ci				

表 22.11　模块显示的菜单表（jos_modules_menu）

字　段	名　称	类　型	整　理	NULL	键　名	默　认	额　外
moduleid	模块 id	int(11)	NULL		PRI	0	
menuid	菜单 id	int(11)	NULL		PRI	0	

（6）系统菜单数据表

说明：系统菜单数据表（jos_menu），菜单表新建任何菜单其实都是在这里增加一条数据。在表中的 menutype 字段记录菜单类型目前系统分为 4 种菜单。

- Mainmenu：主菜单。
- Topmenu：顶菜单。
- Othermenu：其他菜单。
- Usermenu：用户菜单。

表 22.12　系统菜单数据表（jos_menu）

字　段	名　称	类　型	整　理	NULL	键　名	默　认	额外
id	Id	int(11)	NULL		PRI	（NULL）	auto_increment
menutype	菜单类型	varchar(225)	utf8_general_ci	YES	MUL	（NULL）	
name	菜单名称	text	utf8_general_ci	YES		（NULL）	
link	链接	text	utf8_general_ci	YES		（NULL）	
type	类别	varchar(150)	utf8_general_ci				
published	是否发布	tinyint(1)	NULL			0	
parent	父 id	int(11) unsigned	NULL			0	
componenttid	组件 id	int(11) unsigned	NULL		MUL	0	
sublevel	下级菜单	int(11)	NULL	YES		0	
ordering	排序	int(11)	NULL	YES		0	
checked_out	检查	int(11) unsigned	NULL			0	
checked_out_time	检查时间	datetime	NULL			0000-00-00 00:00:00	
access	权限	tinyint(3) unsigned	NULL			0	
utaccess	Ut 权限	tinyint(3) unsigned	NULL			0	
params	参数	text	utf8_general_ci				
lft	位置（左）	int(11) unsigned	NULL			0	
rgt	位置（右）	int(11) unsigned	NULL			0	
home	首页	int(1) unsigned	NULL			0	

（7）用户的权限控制相关表

说明：用户的权限控制相关表包括 jos_core_acl_aro 表、jos_core_acl_aro_groups 表、jos_core_acl_aro_sections 表、jos_core_acl_groups_aro_map 表和 jos_groups 表，这几个表用于存储用户的权限控制信息。该部分涉及的表众多且与本章关联较小，因此不在此做详细的说明。具体 Joomla 的用户权限控制是采用的开源的 phpGACL 权限控制框架，详细信息可以参考“phpGACL”的文档。

（8）用户站内消息系统

说明：用户站内消息系统（com_message），涉及的表有两个，分别是 jos_messages 表和 jos_messages_cfg 表（该表为用户站内消息表的附表，这里不做说明）。

表 22.13　用户站内消息表

字　段	名　称	类　型	整　理	NULL	键　名	默　认	额　外
message_id	消息 id	int(10) unsigned	NULL		PRI	（NULL）	auto_inc-rement
user_id_from	来源用户	int(10) unsigned	NULL			0	
user_id_to	发送用户	int(10) unsigned	NULL			0	
folder_id	文件夹 id	int(10) unsigned	NULL			0	
date_time	日期	datetime	NULL			0000-00-00 00:00:00	
state	消息状态	int(11)	NULL			0	
Priority	优先级	int(1) unsigned	NULL			0	
subject	主题	text	utf8_general_ci				
message	消息内容	text	utf8_general_ci				

（9）模板相关表

说明：模板相关表包括模板风格表（jos_template_styles）。Joomla 可以采用多模板机制，可以为不同的菜单分配不同的模板。

表 22.14　模板风格表

字段	名　称	类　型	整　理	NULL	键名	默　认	额　外
id	id	int(10)	utf8_general_ci	YES	PRI	（NULL）	auto_increment
Templ-ate	模板文件夹	Varchar(50)	utf8_general_ci	YES			
client_id	前台/后台模板标识	tinyint(1)	NULL	YES			
Home	默认当前模板	char(10)	utf8_general_ci	YES			
Title	模板名称	Varchar(255)	utf8_general_ci	YES			
params	参数	text	utf8_general_ci	YES			

22.4　下载与安装

Joomla 是当前最流行的开源 CMS 项目之一。现在介绍如何获得 Joomla 并且正确地设置它，然后通过浏览器使用 Joomla 的安装向导引导用户完成安装。

没有接触过开源软件的读者，可能对如何安装 Joomla 一头雾水，其实 Joomla 的安装非常简便。Joomla 安装分为 3 个步骤：

（1）获取最新的 Joomla 文件包。

（2）将文件包解压到 Web Server 的发布目录中。

（3）通过浏览器使用 Joomla 的安装向导引导用户完成安装。

22.4.1　下载及安装环境需求

目前 Joomla 的最新的稳定版本是 1.6.1，读者可以通过 Joomla 的官方网站（www.joomla.org）获得源文件包，以及最新的更新。但是 Joomla 的官方网站并没有提供中文用户使用的简体中文语言包，读者可以通过国内的站点，下载翻译过的 Joomla 的中文语言包，比如通过下面的地址获得：

```
http://www.joomlagate.com/
```

注意：本章中讲解的内容将以 Joomla 1.6.1 这个版本为准，该版本的相关信息读者可以参考 Joomla 的帮助文档。

Joomla 安装环境要求如表 22.15 所示。

表 22.15　Joomla安装环境要求

软 件 需 求	最 低 版 本	下 载 地 址
PHP	5.2.x	http://www.php.net/
MySQL	4.1.x	http://www.mysql.com/
Apache	2.x	http://www.apache.org
Microsoft IIS	7	http://www.iis.net

注意：Joomla 目前不支持 MySQL6.x 版本。

22.4.2　文件包解压及配置设置

获得 Joomla 的文件包之后，将该文件包解压到 Web Server 的发布目录中。但是在正式安装之前，需要检查几项必要的设置，Joomla 的安装过程需要这些环境的支持。

必要配置除了前面提到的基础开发环境以外，还需要以下支持。

（1）XML 支持：一些新版本 PHP 集成环境中已经内置了 XML 模块，读者可以通过查看 PHP 的该配置信息，了解 XML 配置信息，如图 22.13 所示。

当“XML Support”选项为“active”时，表示 XML 支持已开启。

（2）ZLib 压缩支持：新版本 PHP 集成环境中已经内置了 ZLib 模块，读者可以通过查看 PHP 的配置信息，了解 ZLib 配置信息，如图 22.14 所示。

当“ZLib Support”选项为“enabled”时，表示 ZLib 压缩支持已开启。

（3）全局配置文件（configuration.php）：该文件是 Joomla 的全局配置文件，所以该文件需要有读、写权限，测试环境下建议设置为 775。

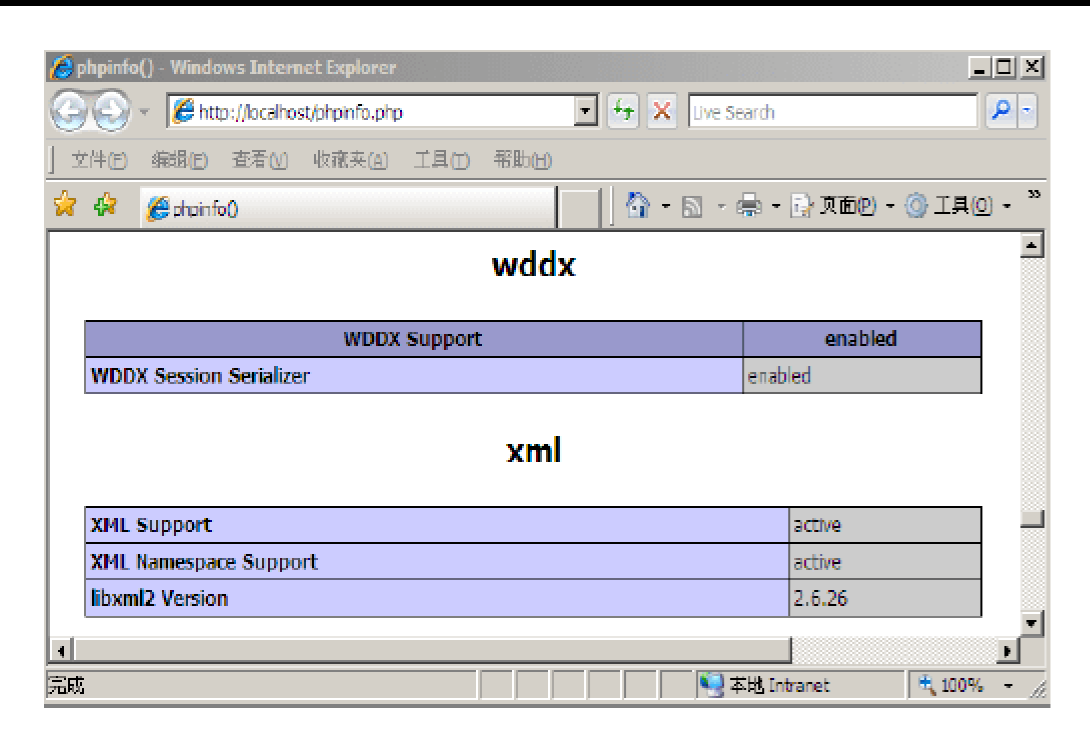


图 22.13　XML 配置信息

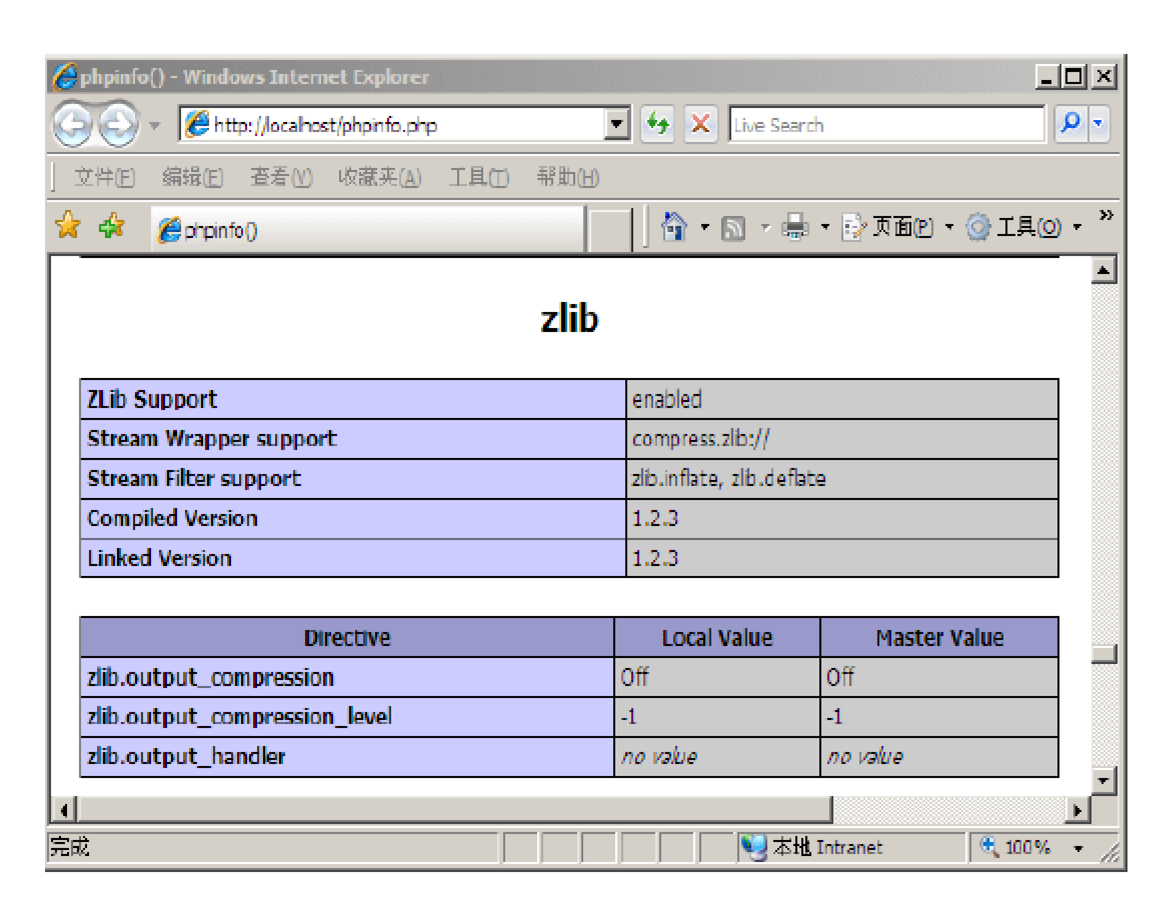


图 22.14　ZLib 配置信息

注意：Joomla 1.6.1 对服务器配置要求提高了，需要 PHP5.2 或以上版本，很多服务器都没有达到 5.2.4 以上的版本，大多在 5.2.0～5.2.3 之间。为了能在服务器上顺利安装 Joomla 1.6.1，对其 PHP 版本的检测修改为 5.2.0。修改两个文件就可以实现。修改保存以后就可以在 5.2.0 或以上的版本运行。

（1）修改“installation/index.php”的 Line：10，将 5.2.4 修改为 5.2.0。

（2）修改“installation\models\setup.php”。这里有两处，分别是 Line：224 和 225，同样将 5.2.4 修改为 5.2.0。

22.4.3　Joomla 的安装向导

当做好前面的准备工作之后，就可以开始安装 Joomla 了。在浏览器中输入 Joomla 在 Web Server 中的发布地址，地址如下：

```
http://localhost/joomla/installation/index.php
```

进入了 Joomla 的安装向导界面，界面如图 22.15 所示。根据安装向导的提示安装过程分为 7 步完成，具体步骤是：

（1）选择语言。

（2）安装前检查。

（3）许可协议。

（4）数据库设定。

（5）FTP 配置。

（6）设置。

（7）完成。



图 22.15　Joomla 安装向导

1．选择语言

在安装向导的右侧“语言选择”项中，列出了 Joomla 的全部语言包。这里采用简体中文作为软件安装语言，选择“简体中文（中国）”选项，然后单击“下一步”按钮，完成选择语言。

2．安装前检查

当选择完安装语言后就进入“安装前检查”步骤，该步骤是安装过程中一个很重要的部分，由检测页面可以看到最低的系统要求是否满足。如果这些参数是红色的（不匹配），那么就需要对相应的设置作出调整（如升级 PHP 版本），如图 22.16 所示。

注意：查看 configuration.php 是否可写，即前文提到的该文件的权限问题。

在“推荐设置”选项是参考选项，如果不匹配也可以正常安装 Joomla，但是可能会出现一些功能上和安全上的问题。确认无误后单击“下一步”按钮，进入下一步的操作。

3．许可协议

Joomla 是 GUN/GPL 协议下的软件，协议内容是 Joomla 软件的具体授权规定，这里建议保留页脚的链接，许可协议如图 22.17 所示。阅读许可协议后，单击“下一步”按钮，进入下一步“数据库设定”的操作。

4．数据库设定

“数据库设定”是安装过程中最重要的一个步骤，主要是配置数据库用到的基本信息。默认选择数据库类型是 MySQLi，由于使用的环境是虚拟机，所以主机名设定为 localhost，

用户名和密码是数据库的连接账号。

如果数据库名称已经存在，需要重新安装。正常情况下，如果要在一个数据库里安装多个 Joomla，那么我们建议使用 Jos 做为数据库表前缀比较好。



图 22.16　安装前检查



图 22.17　许可协议

🔔注意：Jos 为默认表前缀。

确认设置完毕，这时候可以单击“下一步”按钮，安装好 MySQL 数据库，接着再进行下一步的安装，如图 22.18 所示。



图 22.18　数据库设定

5．FTP配置

Joomla 默认不启动用 FTP 功能，如果需要使用该功能则需要填写相关账号信息，如图 22.19 所示。当设定好 FTP 账号信息之后，通过 Joomla 后台自带的 FTP 功能，可以将用户本地文件上传到指定的 FTP 服务器。配置好之后单击“下一步”按钮，完成配置 FTP 的操作。



图 22.19　FTP 配置

6．设置

该步骤是设置网站的主要设置，需要设置的选项包括“网站名称”、“管理员账号信息”和“安装网站示范数据”。如果需要安装示范数据，单击“安装示范数据”按钮，如图 22.20 所示。



图 22.20　Joomla 主要设置

7．完成

经过前面的步骤 Joomla 的设置已经完成，提示信息如图 22.21 所示。这里需要注意的

是，设置完成之后需要删除 installation 目录。



图 22.21　设置完成

22.5　后台基础管理

经过前面的介绍 Joomla 的准备知识，相信读者已经对 Joomla 的组成和结构有了一个清晰的认识。从本节开始正式使用和操作 Joomla，本节首先介绍 Joomla 的权限角色划分，然后再逐步讲解 Joomla 后台的基础管理功能。

22.5.1　权限角色划分

通过 Joomla 的安装向导安装系统之后，用户会得到两个访问地址。

（1）公共站点访问地址（前台）：http://localhost/Joomla/。

（2）管理站点地址（后台）：http://localhost/Joomla/administrator/。

Joomla 的规范中，规定前台用户分为 4 种基本类型，每种类型（角色）的功能和具体权限如下所示。

- ❑ 注册用户组：这个类型（组）是网站中最基本的类型，使用频率也最高。通常应用在需要注册用户才能获得内容的权限设计中。
- ❑ 作者角色（Author）：该角色拥有注册用户角色的全部权限，可以新建和修改自身文章。
- ❑ 编辑角色（Editor）：该角色拥有作者角色的全部权限（编辑属于作者的一种情况），可以修改其他用户的文章。
- ❑ 发布者角色（Publisher）：拥有编辑角色的全部权限，并且可以发布文章（在网站前台显示）。

同样，Joomla 的规范中也规定了后台用户的 3 种基本类型，每种类型（角色）的功能和具体权限如下所示。

- ❑ 管理员角色：该角色通过网站的管理后台管理网站，拥有基本的一些操作权限（如

新建文章等）。这里需要注意的是，管理员角色是基本管理者权限，不能进行其他高级别的操作（如安装模块、组件、修改后台全局设置，提升其他管理员权限等）。

- 高级管理员角色：该角色拥有管理员角色的全部权限，拥有安装模块、组件的权限，但是依然不能修改全局设置和提升其他管理员权限。
- 超级管理员角色：该角色拥有高级管理员角色的全部权限，超级管理员角色可以创建另一个超级管理员，即提升其他管理员角色的权限。

22.5.2　后台概述

前面介绍了用户角色的划分及各角色的权限，Joomla 后台的各项基础管理就是以后台角色权限划分为基础设置的。Joomla 后台的功能是有效地管理网站的基础信息和内容，由于网站本身的定制化和扩展性的需要，就需要一套规则和管理功能来管理基础后台的基础设置。下面逐一介绍这些基础管理功能，它们是后面内容管理的基础。

（1）首先打开浏览器输入后台管理地址：

```
http://localhost/Joomla/administrator/
```

显示后台管理的登录界面，如图 22.22 所示。



图 22.22　后台管理登录

（2）在登录界面中有 3 个选项，即账号、密码和语言。前面两项是登录的口令，读者在前面安装过程中已经进行了相应的设置，语言选择项中选择“简体中文”，登录成功后管理后台的界面将是简体中文显示。

注意： 由于本章使用的 Joomla 是汉化版，因此在选择语言选项时，需要在安装过程中的“设置”步骤中选择安装简体中文的示范数据，以保证显示简体中文界面。

（3）登录成功后进入后台管理界面如图 22.23 所示。从图中可以看到，后台管理界面按照功能划分为 4 个部分，分别是管理菜单导航栏（区域 1）、状态及工具栏（区域 2）、控制面板框（区域 3）、活动状态栏（区域 4）。

- 管理菜单导航栏：管理菜单导航栏的作用是提供便捷的操作管理导航，包括网站、用户、菜单、内容、组件、扩展和说明几项内容，该选项都有具体的管理内容页面。
- 状态及工具栏：该部分显示一些基本的状态信息（如登录状态等）、站内短信息、打开网站前台选项（页面预览功能）和退出登录功能。

- 控制面板框：控制面板框功能和管理菜单导航中的内容是重叠的，其意义是可以方便用户快速地找到常用的功能。
- 活动状态栏：活动状态栏部分可以帮助用户方便地了解网站的基础运行数据，便于管理员对网站进行有针对性的管理。



图 22.23　后台管理界面

22.5.3　语言管理

Joomla 是一个开源的国际项目，因此它默认的系统语言是英文，为了方便其他国家的使用者参与到 Joomla 的开发中来，Joomla 系统本身提供了多种语言包接口。通过这些接口可以扩展 Joomla 的语言。下面介绍 Joomla 的第一个基础管理功能语言管理，Joomla 的语言包按照使用方式分为前台语言包和后台语言包。需要注意的是，前台语言包和后台语言包分属不同的系统，不能共用。

现在以安装和管理简体中文语言包为例，介绍 Joomla 对后台语言的管理。

1．语言包下载

首先打开 Joomla 官方网站（http://joomlacode.org/gf/project/jtranslation/frs/?action=FrsReleaseBrowse&frs_package_id=2468），下载对应版本的前台和后台的简体中文语言包。

下载界面如图 22.24 所示。其中，Joomla1.6.0_Simplified_Chinese_Site 是 Joomla 前台简体中文语言包，下载到本地后的文件名为 zh-CN_site.zip；Joomla1.6.0_Simplified_Chinese_admin 是 Joomla 后台简体中文语言包，下载到本地后的文件名为 zh-CN_administrator.zip。

发行名称	发行日期	Filename
Joomla1.6.x_Installation_Language	2011-02-21 10:00:00-06	zh-CN.zip
Joomla1.6.1_Simplified_Chinese_Package	2011-03-14 11:00:00-05	zh-CN.zip
Joomla1.6.0_Simplified_Chinese_Site	2011-02-09 10:00:00-06	zh-CN_site.zip
Joomla1.6.0_Simplified_Chinese_admin	2011-02-09 10:00:00-06	admin_zh-CN.zip

图 22.24　语言包下载

2．语言包上传及文件说明

由于 Joomla 的前台语言和后台语言是相互独立的，分属不同的系统，文件分别处于不同的目录中，因此将下载的语言包要分别上传到对应的目录中。

在第 22.2 节中为读者介绍过 Joomla 的目录含义。其中，Joomla 的前台语言包目录位置为./language/，打开该目录新建 zh-CN 文件夹，然后将下载的前台中文语言包 zh-CN_site.zip 解压后上传至该目录中。解压后该目录如图 22.25 所示。

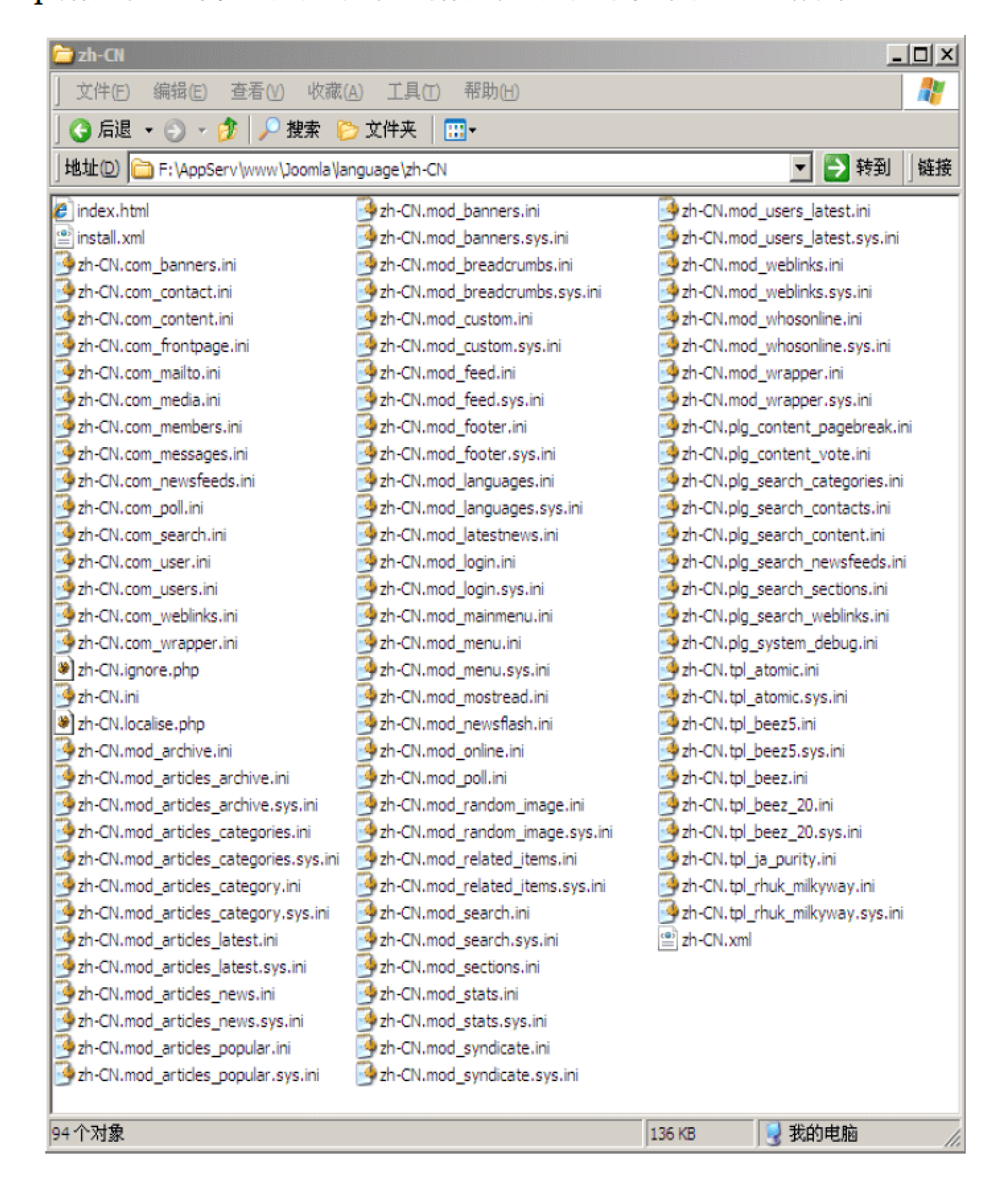


图 22.25　前台语言包目录

由于语言包文件众多，这里为读者介绍几个重要的文件的含义。

（1）zh-CN.ini 文件

该文件是语言包的核心文件，定义了后台页面元素对应的语言翻译。以编辑文章相关的信息为例，说明代码部分如下：

```
JGLOBAL_ARTICLE_MUST_HAVE_TEXT="文章必须要有内容"
JGLOBAL_ARTICLES="文章"
JGLOBAL_AUTH_ACCESS_DENIED="访问权限被拒绝"
JGLOBAL_AUTH_ACCESS_GRANTED="允许访问权限"
JGLOBAL_AUTH_BIND_FAILED="无法绑定到 LDAP 服务器"
JGLOBAL_AUTH_CANCEL="认证被取消"
JGLOBAL_AUTH_EMPTY_PASS_NOT_ALLOWED="不允许空白密码"
JGLOBAL_AUTH_FAIL="验证失败"
```

```
JGLOBAL_AUTH_FAILED="验证失败： %s"
JGLOBAL_AUTH_INCORRECT="不正确的账号/密码"
JGLOBAL_AUTH_INVALID_PASS="无效的密码"
JGLOBAL_AUTH_NO_BIND="无法绑定到 LDAP"
JGLOBAL_AUTH_NO_CONNECT="无法连接到 LDAP 服务器"
JGLOBAL_AUTH_NO_REDIRECT="无法自定义链接到服务器： %s"
JGLOBAL_AUTH_NO_USER="用户不存在"
JGLOBAL_AUTH_NOT_CREATE_DIR="无法创建 FileStore 的目录%s.请检查权限是否有效."
JGLOBAL_AUTH_PASS_BLANK="LADP 不能有空白密码"
JGLOBAL_AUTH_UNKNOWN_ACCESS_DENIED="结果不详.访问权限被拒绝"
JGLOBAL_AUTH_USER_BLACKLISTED="用户被列入黑名单"
JGLOBAL_AUTH_USER_NOT_FOUND="找不到用户"
JGLOBAL_AUTO="自动"
JGLOBAL_CATEGORY_NOT_FOUND="类别未找到"
JGLOBAL_CLICK_TO_SORT_THIS_COLUMN="点按可以依这个字段来排序"
JGLOBAL_CREATED_DATE_ON="创建于 %s"
JGLOBAL_DESCRIPTION="说明"
JGLOBAL_DISPLAY_NUM="显示数量"
JGLOBAL_EDIT="编辑"
JGLOBAL_EMAIL="Email"
JGLOBAL_FIELD_CREATED_BY_ALIAS_DESC="使用另一个作者名称来显示"
JGLOBAL_FIELD_CREATED_BY_ALIAS_LABEL="作者别名"
JGLOBAL_FIELD_FEATURED_DESC="指定文章到推荐部落格布局中"
JGLOBAL_FIELD_FEATURED_LABEL="推荐"
JGLOBAL_FIELD_PUBLISH_DOWN_DESC="可选的停止发布日期"
JGLOBAL_FIELD_PUBLISH_DOWN_LABEL="结束发布"
JGLOBAL_FIELD_PUBLISH_UP_DESC="可选的开始发布日期"
JGLOBAL_FIELD_PUBLISH_UP_LABEL="开始发布"
JGLOBAL_FILTER_BUTTON="过滤"
JGLOBAL_FILTER_LABEL="过滤"
JGLOBAL_FULL_TEXT="全文"
JGLOBAL_GT="&gt;"
JGLOBAL_HITS="点击数"
JGLOBAL_ICON_SEP="|"
JGLOBAL_INHERIT="继承"
JGLOBAL_INTRO_TEXT="摘要文字"
JGLOBAL_LEFT="左"
JGLOBAL_LT="&lt;"
JGLOBAL_NEWITEMSLAST_DESC="新项目默认排在最后.当这项目保存之后, 顺序可以再更改."
JGLOBAL_NUM="数"
JGLOBAL_PASSWORD="密码"
JGLOBAL_PRINT="打印"
JGLOBAL_RECORD_NUMBER="记录编号：%d"
JGLOBAL_REMEMBER_ME="记得我"
JGLOBAL_RESOURCE_NOT_FOUND="资源无法找到"
JGLOBAL_RIGHT="右"
JGLOBAL_STATE="状态"
JGLOBAL_SUBCATEGORIES="子类别"
JGLOBAL_TITLE="标题"
JGLOBAL_USE_GLOBAL="使用全站设定"
JGLOBAL_USERNAME="账号"
JGLOBAL_VALIDATION_FORM_FAILED="无效的格式"
JGLOBAL_YOU_MUST_LOGIN_FIRST="请先登录"
```

【代码解读】

默认英文名称后，对应的就是中文的显示内容，这些对应信息在使用中非常有效，可以帮助使用者快速定位错误内容。

（2）zh-CN.xml 文件

该文件默认的配置文件。具体地说，就是当用户进入后台登录界面时，语言下拉框中加载的选项，如图 22.26 所示。



图 22.26　登录界面语言选择

在编辑器中打开该文件代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<metafile version="1.6" client="site">
    <tag>zh-CN</tag>
    <name>简体中文</name>
    <description>Joomla 1.6 前台简体中文（Simplified Chinese）语言文件
    </description>
    <version>1.6.0</version>
    <creationDate>2010-02-09</creationDate>
    <author>Joomla 之门</author>
    <authorEmail>joomlagate@gmail.com</authorEmail>
    <authorUrl>http://www.joomlagate.com</authorUrl>
    <copyright>Copyright (C) 2005 - 2011 Open Source Matters &amp; Joomla.fr.
    All rights reserved.</copyright>
    <license>GNU General Public License version 2 or later; see LICENSE.
    txt</license>
    <metadata>
        <name>简体中文</name>
        <tag>zh-CN</tag>
        <rtl>0</rtl>
        <locale>zh CN.utf8, zh, zho, zhi, zh CN, zho CN, zhi CN,simplified
        _chinese</locale>
        <firstDay>1</firstDay>
    </metadata>
    <params />
</metafile>
```

【代码解读】

上述代码是一个标准的 XML 格式的配置文件，前部定义了该配置文件的基础信息（如版本信息、创建日期、描述等）。后面是配置信息的主体部分，即<metadata></metadata>

标签之间的部分。其中，name 标签是提供后台选择的语言包的名字、tag 标签是语言包显示标签、locale 是该语言包的应用场合。

后台语言包目录位置为./administrator/language/，同样在该目录新建 zh-CN 文件夹，然后将下载的后台中文语言包 zh-CN_administrator.zip 解压后上传至该目录中。后台语言包的结构和内容与前台语言包相似，读者可以参看前面的介绍，这里不再赘述。

3．语言包安装

在前面已经将前台和后台的语言包上传到指定的目录中了，下面介绍如何通过 Joomla 后台的扩展管理功能完成语言包的安装。

（1）登录 Joomla 管理后台后，选择“扩展”|“扩展管理”命令，如图 22.27 所示。



图 22.27　扩展管理

（2）打开“扩展管理”设置页面后，单击“探索”选项，然后单击右上方工具栏中的“探索”按钮，如图 22.28 所示。



图 22.28　文件检查

（3）“探索”完毕后，会提示已经检测出了之前上传的 Joomla 前台和后台的简体中文语言包，勾选这两个语言包选项，然后单击工具栏中的“安装”按钮完成语言包安装。

4．语言管理

安装成功后，通过语言管理后台将简体中文语言包设置为网站前台和后台的默认语言包。单击菜单栏中的“扩展”选项，然后选择“语言管理”，进入语言管理后台，如图 22.29 所示。



图 22.29　语言管理后台

（1）前台语言选择：在右侧“过滤器位置”的下拉列表框中选择“网站”选项，然后勾选“简体中文”选项，单击工具栏中的“默认”按钮，将简体中文设置为 Joomla 网站前台的默认语言。

（2）后台语言选择：在右侧“过滤器位置”的下拉列表框中选择“后台”选项，勾选“简体中文”选项，单击工具栏中的“默认”按钮，将简体中文设置为 Joomla 网站后台的默认语言。

22.5.4　媒体文件管理

现在网站的内容呈现形式越来越丰富，常见的像图片轮播、音频、Flash 动画等媒体文件，在网站中引入媒体文件主要是为了丰富网站的形式。Joomla 的媒体文件管理功能主要是针对前台展示页面所用到的媒体文件，除了前面提到的以外，还包括 CSS 样式表文件和 JavaScript 脚本文件等。通过媒体文件管理后台，可以对这些文件进行统一、有效地管理。下面逐一介绍媒体文件目录创建、媒体文件上传和文件本身的管理（删除操作等）。

1．媒体目录管理

在后台管理的“控制面板”区域单击“媒体管理”图标，进入媒体管理的管理界面，如图 22.30 所示。

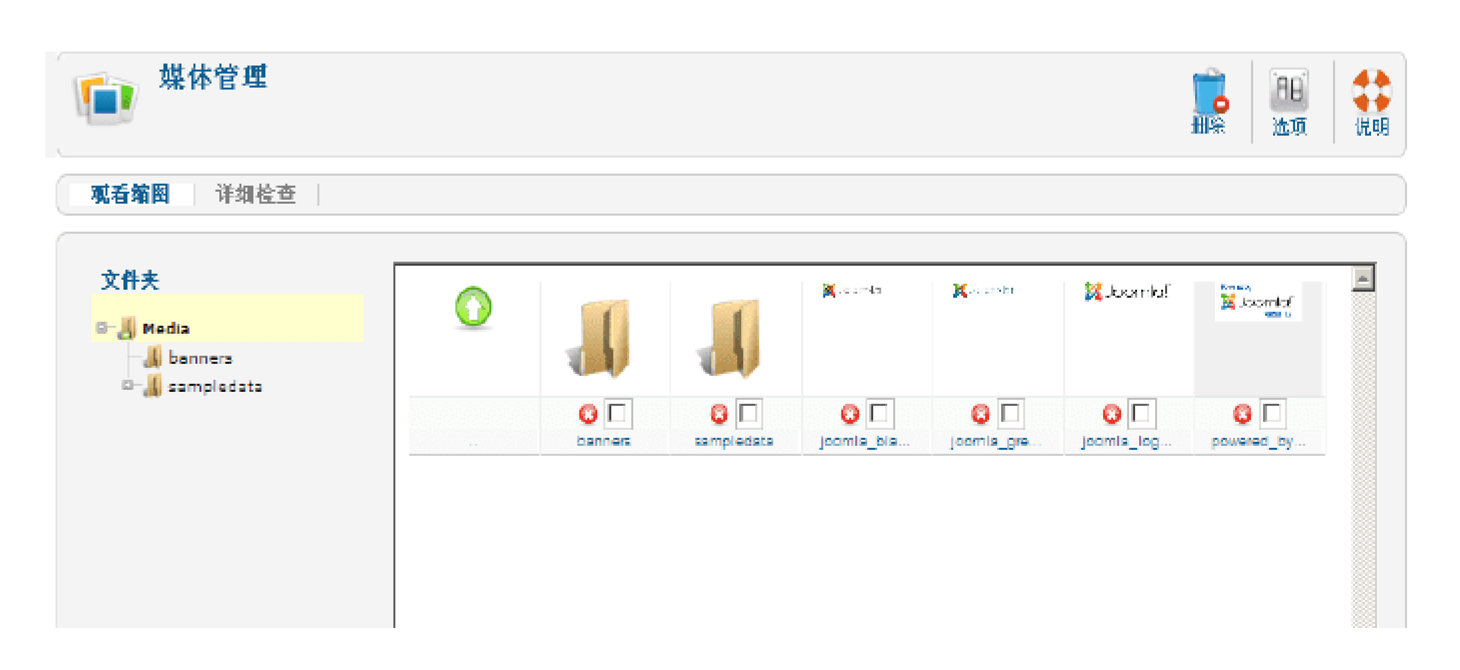


图 22.30　媒体管理缩略图模式

用户首先会看到已有媒体的列表，这里系统提供了两种方式即“缩略图”方式和“详情”方式。系统默认的是缩略图，读者可以对比“详情”方式比较两种方式各自的特点。

如图 22.31 所示。

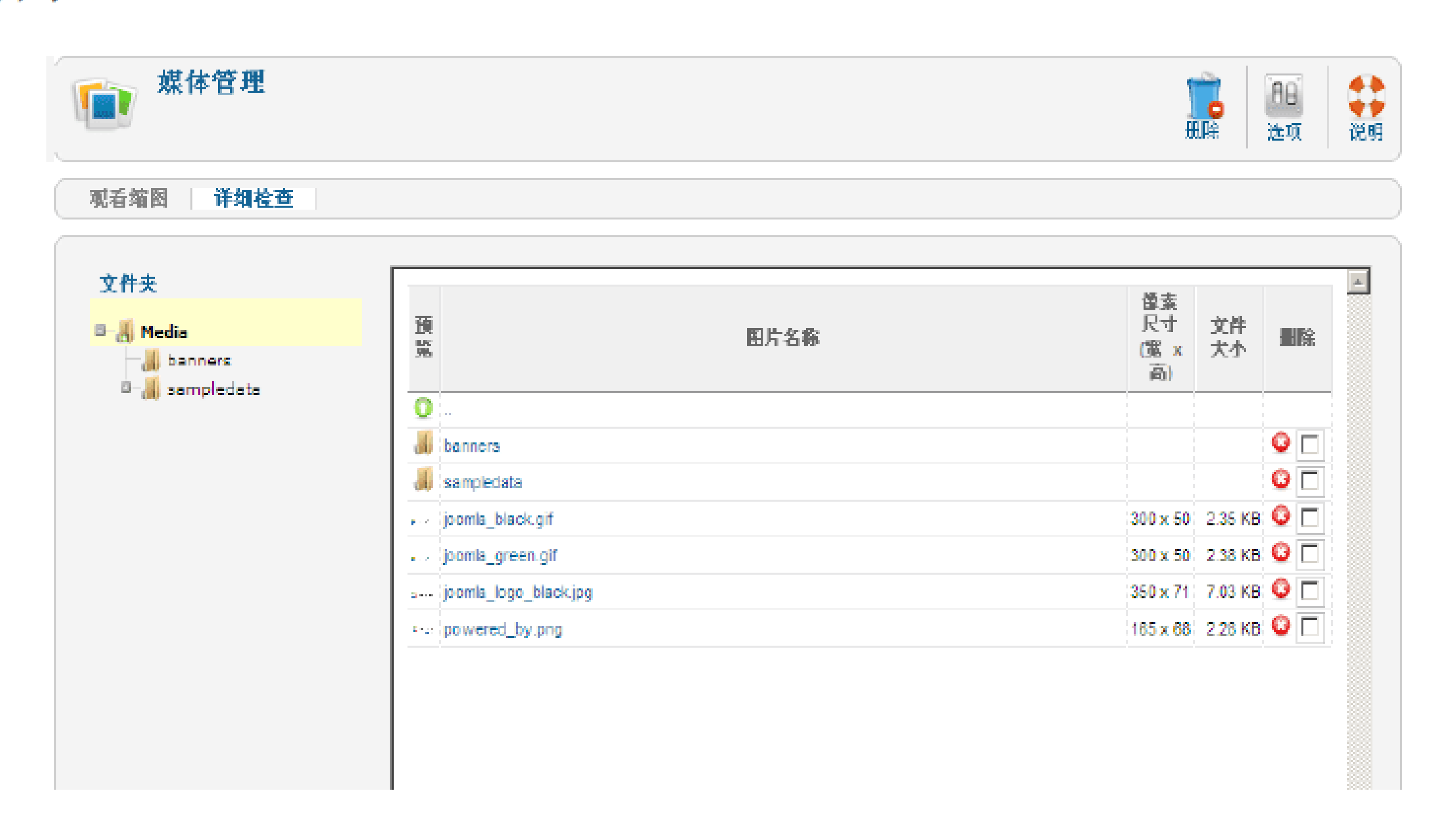


图 22.31　媒体管理详情模式

如果所有的媒体文件都不做归档直接放在根目录下，那将是极为混乱的，也难于管理。因此，需要为不同用途的媒体文件创建相应的目录，这样管理起来简洁、明了。下面介绍如何创建媒体目录。

首先在“文件”区域的文本框中，输入要创建的文件夹名字，这里输入“test”作为新建文件夹的名字。然后单击“文件”区域的“新建文件夹”按钮，完成创建媒体目录。如图 22.32 所示，在左侧目录树中已经显示出已经创建好的媒体目录。

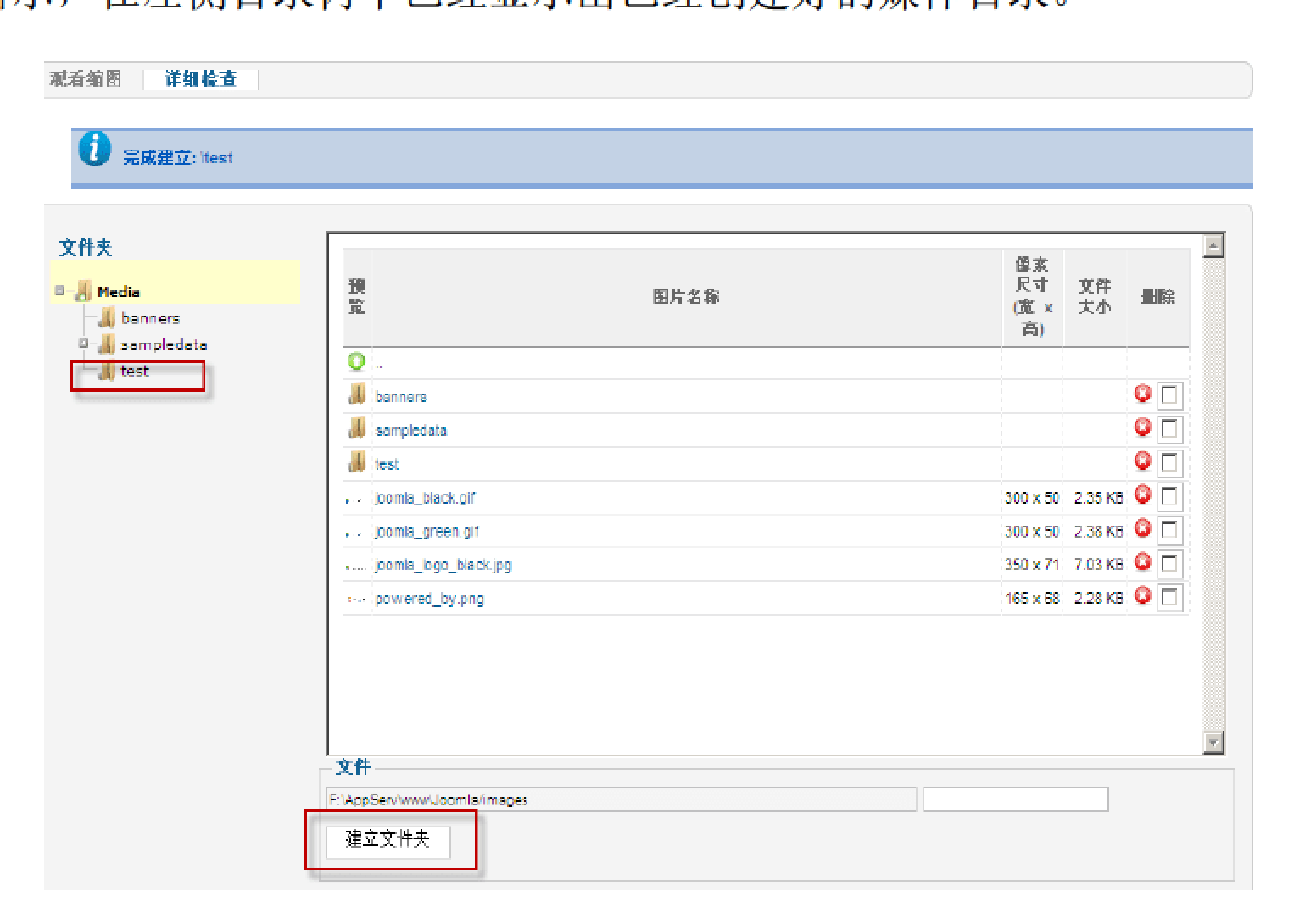


图 22.32　创建媒体目录

2．媒体文件管理

下面介绍如何上传一个媒体文件到系统中，媒体管理后台提供了文件上传的功能，如图 22.33 所示。选择文件后单击“开始上传”按钮，媒体文件将被上传到指定的媒体目录中去，这里设置的目录地址如下：

```
F:\AppServ\www\Joomla/images
```

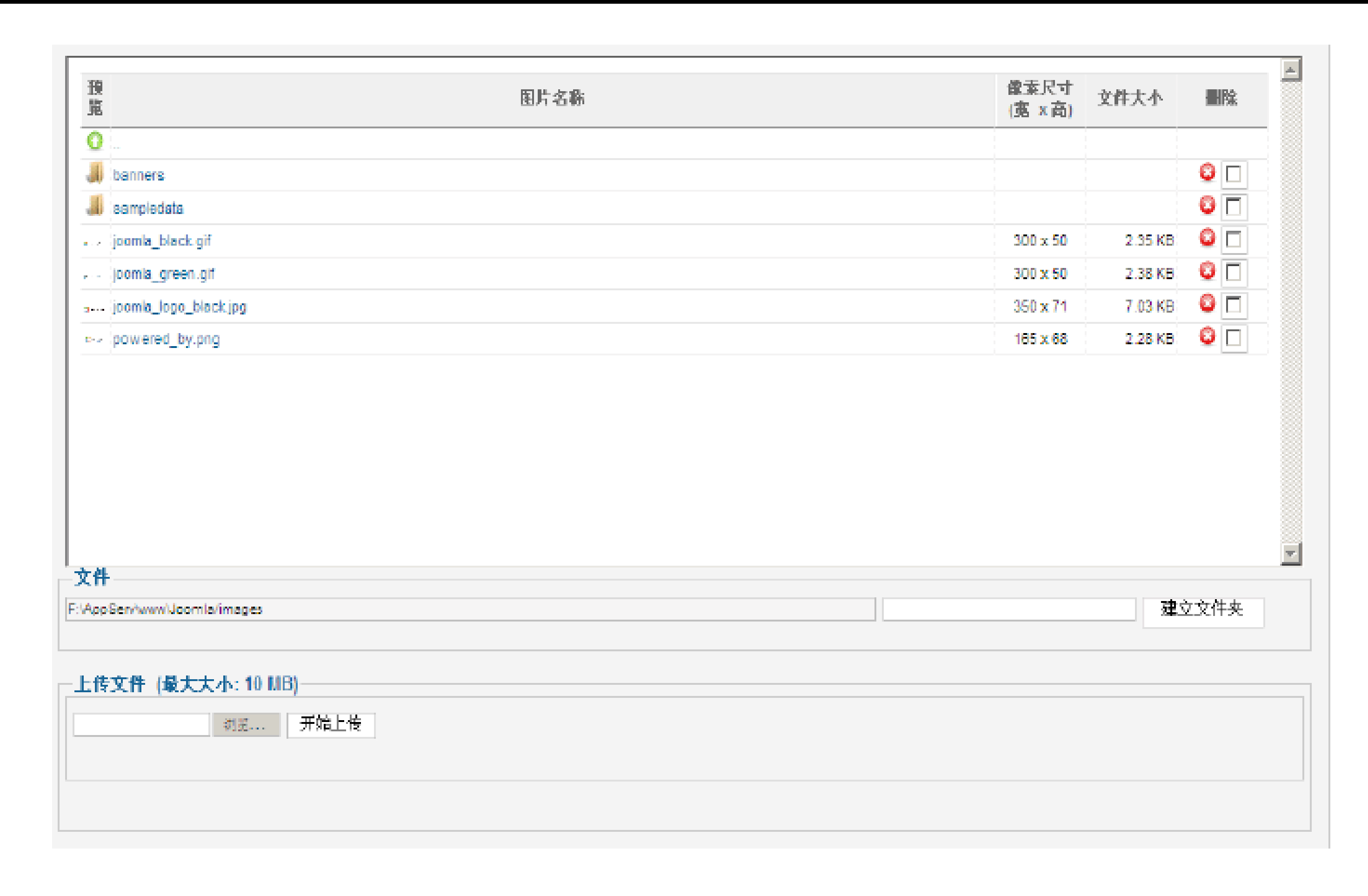


图 22.33　媒体文件上传

上传成功后，会在媒体文件列表中显示已经上传成功的媒体文件，这里上传一个文件名为 test.jpg 的图片媒体文件，如图 22.34 所示已经成功上传在媒体文件列表中，也可以找到刚刚上传的测试媒体文件。

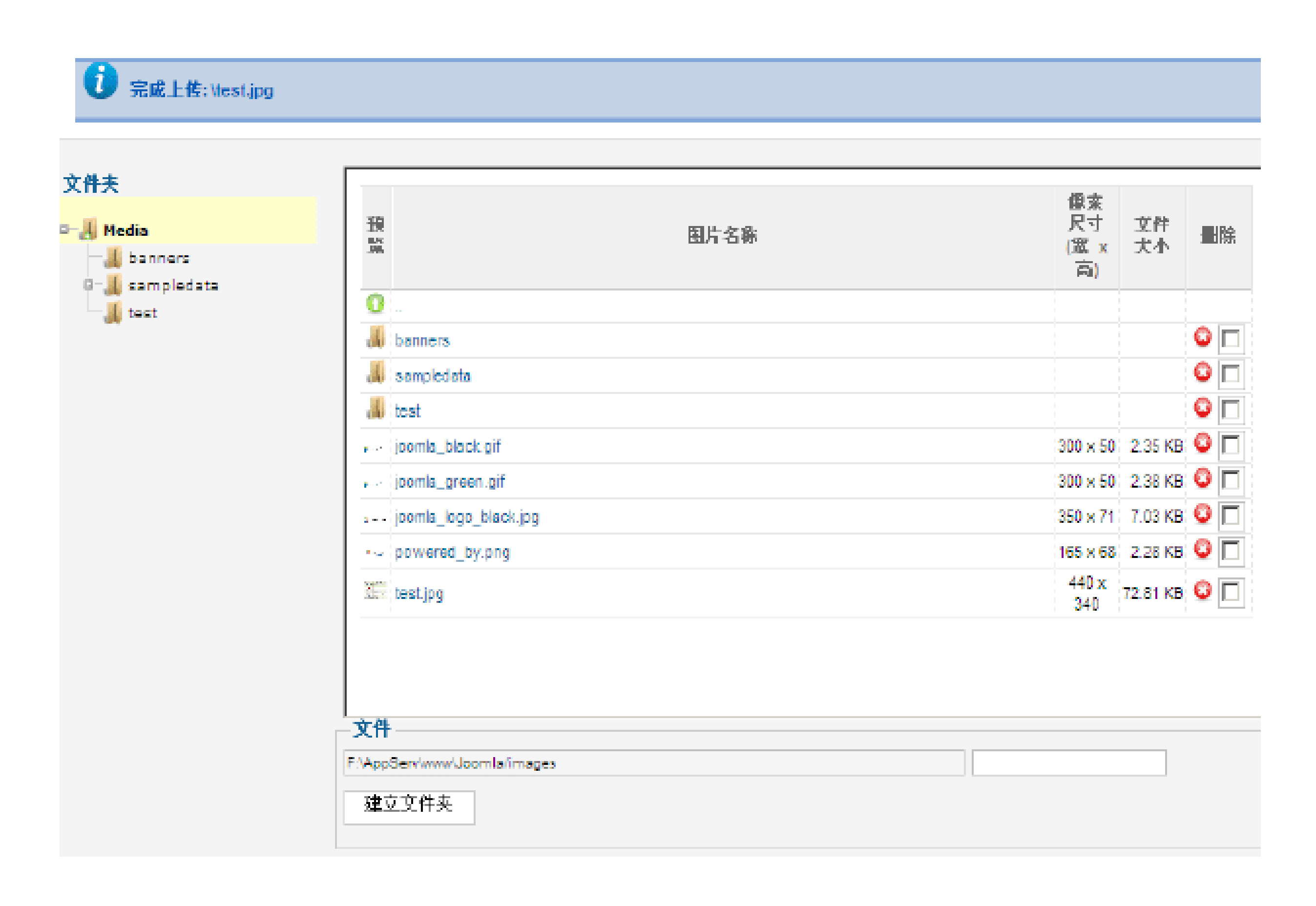


图 22.34　媒体文件上传成功

媒体管理后台提供了两个管理媒体文件的功能，即图片文件预览功能和已上传图片删除功能。

首先单击列表中的文件图标，效果如图 22.35 所示，可以看到图片文件的缩略图预览。

媒体文件的删除功能十分简便，勾选要删除的媒体文件图片，然后在右侧单击删除按钮图片，完成删除操作。

22.5.5　全局设置

在前面介绍 Joomla 使用向导安装时，已经对系统做了初始化的设置。但是有时需要查看这些设置或者监控系统的运行情况，并做出一些调整，这时可以通过管理后台提供的“全局设置”功能进行查看或是调整。

首先单击“控制面板”中的“全局设置”图片，进入全局设置管理界面，如图 22.36

所示。从图中可以看到“全局设置”栏目分为 4 个子管理栏目，各自的功能如下。



图 22.35　图片媒体文件预览

- 网站：“全局设置”栏目的默认子栏目，显示网站的基础信息（如网站名称等）。
- 系统：查看基础的系统配置信息（如日志文件路径设置、缓存策略、Session 生存周期等）。
- 服务器：包含服务器、FTP 设置、数据库设置和邮件的基础设置信息。
- 权限：查看和修改基本的用户角色和管理员角色权限。



图 22.36　全局设置

22.6　后台内容管理

Joomla 作为一个典型的内容管理系统（CMS），其核心的功能就是网站内容的组织。如何组织、展示站点上的内容，内容管理系统（CMS）实现的策略都不同。在 Joomla 中，

这些分散的内容会被分成两个层次，即单元和分类。通过单元和分类的组合完成网站基本内容的组织。

本节主要为读者介绍如何通过前面设置好的 Joomla 的后台完成菜单管理及导航，如何进行单元管理、类别管理和文章管理。本章中会涉及一些 Joomla 特有的概念语言，读者在学习时注意区别。

22.6.1　菜单管理

菜单管理是内容管理部分的基础和前提条件，后续的内容管理都是建立在它的基础之上的。菜单的创建与管理也在一定程度上反映了网站管理者对网站规划的总体思路，因此网站内容建设与管理从创建菜单开始。

在前面提到菜单是内容管理的基础，如类别（category）或者单独的文章（article）等都需要通过直接或者间接方式与菜单（menu）建立关联，这样才能在网站中被用户访问到。

按照与菜单关联方式的不同可以将菜单分为直接菜单（Direct menu）和分类菜单（Category menu）。下面通过具体操作的例子来理解两种菜单的区别，并带读者学习菜单的基本操作。

1．直接菜单（Direct menu）

下面在控制面板中选择“菜单管理”|“新建”命令创建一个“菜单项”（menu item），如图 22.37 所示。该菜单的作用通常是关联到一篇未分类的文章。



图 22.37　菜单管理

进入增加菜单栏目后，如图 22.38 所示。这里需要填 3 个项目内容，即标题（必填）、菜单类型（必填）和说明（可选）。

输入相应信息后单击“保存”按钮，完成新增菜单操作。

2．分类菜单（Category menu）

下面在控制面板中选择“菜单管理”选项，单击“菜单项目”链接，如图 22.39 所示，可以看到安装的测试数据中包含已有的菜单项目。



图 22.38　增加菜单



图 22.39　菜单项目列表

然后单击“新增”按钮进入“新的菜单项目”界面，如图 22.40 所示。这里需要注意的是，有 3 项必填内容，即菜单项目类型、菜单标题和菜单位置。其余各项为可选项目。



图 22.40　新增菜单项目

输入相应信息后单击“保存”按钮，完成新增菜单项目操作。

22.6.2　类别管理

在 Joomla 1.6.1 这个版本中取消了之前版本使用的“单元+类别”的管理模式，而是将类别管理划分为内容管理的子项目之一，这样逻辑层次更为清晰、管理更为便捷。类别管理的核心内容是管理和设置类别，同时设定类别之间的从属关系，以及对应的用户组角色的权限。

下面介绍几类常见的类别管理操作。

1．添加类别项目

初始化的 Joomla 后台是没有默认的类别的(为了便于用户理解本章已安装测试数据)，用户单击控制面板中的类别管理图标，进入类别管理界面，如图 22.41 所示。



图 22.41　类别管理

在管理界面右侧的状态栏图标中，包含了类别管理涉及的基本操作，这些操作包括新建、编辑、发布（停止）等。下面首先介绍如何新增类别。

单击类别管理界面中的“新增”图标进入“新建文件类别”界面。需要设置的内容包括两个部分，即详情部分和类别权限。其中详情部分包括该类别的基础属性信息，如图 22.42 所示，具体内容如下

- 标题（必选项）：新增类别的名称。
- 别名（可选项）：容易记忆的名称。
- 主要（可选项）：设定该类别的从属关系。
- 状态（可选项）：包含 4 个状态，即发布、停止发布、存档、丢到回收站。
- 访问（可选项）：设置访问级别（前台用户浏览权限）。
- 权限（可选项）：权限设置。
- 语言（可选项）：语言选项。
- 说明（可选项）：类别描述文字。

详情

标题 *

别名

主要　- 无上层 -

状态　发布

访问　Public

权限　设置权限

语言　所有

ID　0

说明

Styles　Format

Path:

文章　图片　切换编辑器

图 22.42　新建类别详情

类别权限部分的功能是管理该用户群组的权限设置，如图 22.43 所示。该类别通过对前台用户角色的每种基本操作设定权限的规则，实现对群组的设置。

基本的操作说明如下。

- 建立：建立类别权限。
- 删除：删除指定类别权限。
- 编辑：修改编辑类别权限。
- 编辑状态：修改编辑状态权限。
- 编辑所有者：编辑所有者权限。



类别权限

管理该用户群组的权限设置

Public

操作	选择新的设置[1]	计算出的设置结果[2]
建立	继承	不允许
删除	继承	不允许
编辑	继承	不允许
编辑状态	继承	不允许
编辑拥有者	继承	不允许

|– Manager

图 22.43　新建类别权限设置

设定好相应内容后单击“保存”按钮，完成新建类别的操作。该功能也是类别管理中最核心的功能。

2．编辑已有类别项目

由于在前面 Joomla 的向导安装部分已经安装了测试数据，下面通过测试数据的例子来介绍如何编辑一个已有的类别。

首先回到类别管理主界面，然后勾选需要编辑的类别，单击“编辑”按钮，如图 22.44 所示。



图 22.44　选择编辑已有类别

进入“编辑文章类别”界面，该界面的布局同新建类别界面完全相同，项目内容也完全相同，如图 22.45 所示。在修改时要注意类别的从属关系和权限设置问题，因为待修改类别可能与其他类别有从属的连带关系。



图 22.45　编辑文章类别界面

相应项目的含义在介绍新建类别的部分已经介绍过了，这里不再赘述。修改完毕后单击“保存”按钮图标，完成类别的修改。

3．其他操作

类别管理的基本操作除了前面提到过的“新建类别”、“编辑类别”操作之外，经常用到的还包括删除操作（回收站）和发布（取消发布）。

在类别管理主界面中，选择需要删除的项目，然后单击“回收站”选项，完成删除操

作，如图 22.46 所示。

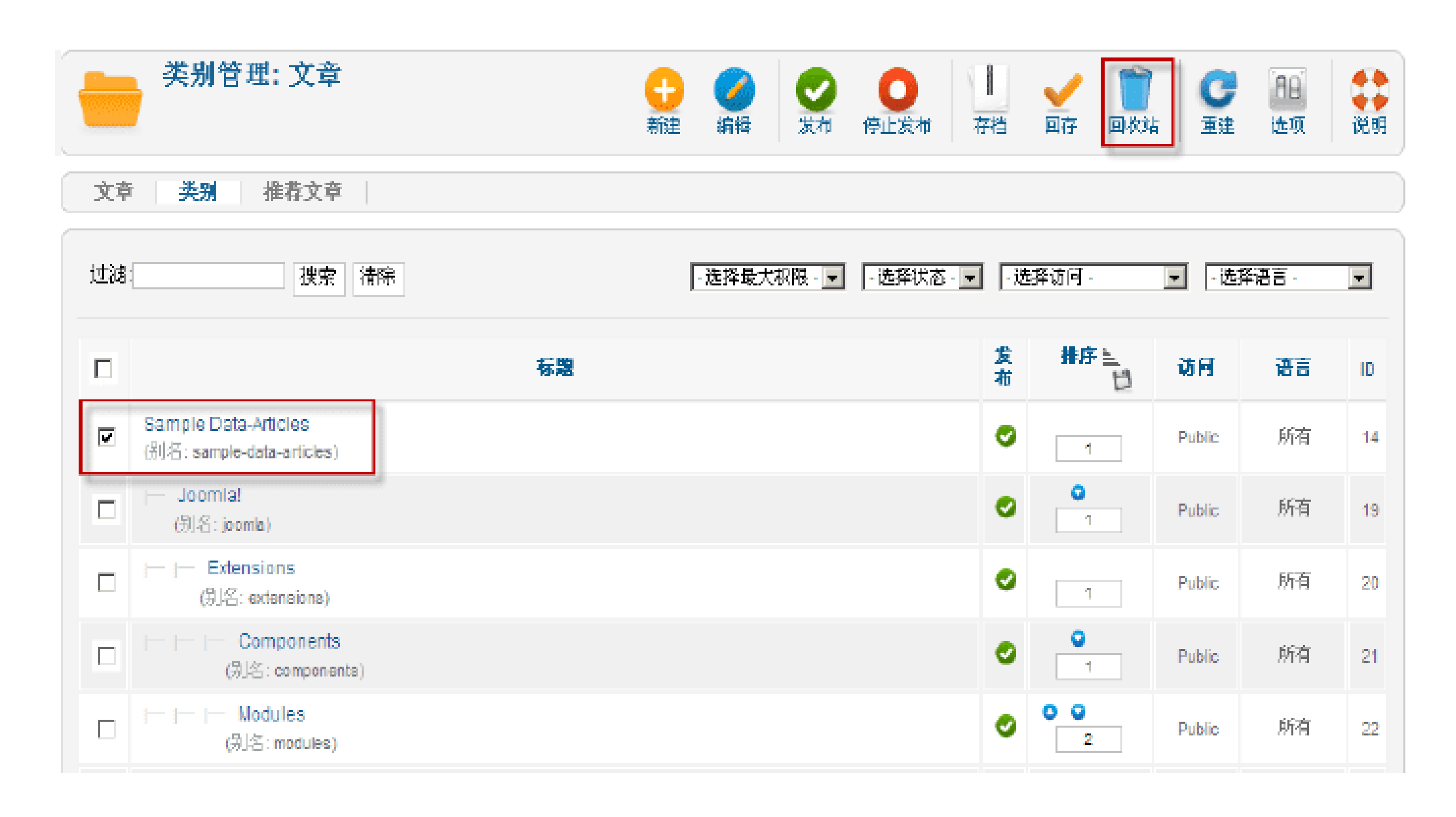


图 22.46　删除指定类别

发布（取消发布）的操作方式与删除操作类似，勾选需要发布（取消）的项目，然后单击“发布”或者“取消发布”选项，完成发布（取消发布）操作。

22.6.3　文章管理

内容管理系统的核心作用就是对内容做有效的管理，在 Joomla 的管理后台中，文章管理是对内容管理的最直接地体现。这里指的文章可以是具体的某一篇文章，也可以是广义上的文章（如一个扩展页面等）。

1．新文章建立

首先介绍内容是如何通过文章管理后台，录入到内容管理系统中的。在后台控制面板中单击“新增文章”选项，进入“新增文章”界面。该界面包含两个基本的管理区域，即新文章区域和文章权限区域。下面介绍“新文章”区域的内容项。如图 22.47 所示，从该区域包含 9 个项目，具体含义如下。

- 标题（必选项）：文章标题。
- 别名（可选项）：容易记忆的名字。
- 类别（必选项）：所属类别。
- 状态（可选项）：包含 4 个状态，即发布、停止发布、存档、丢到回收站。
- 访问（可选项）：设置访问级别（前台用户浏览权限）。
- 权限（可选项）：权限设置。
- 推荐（可选项）：是否可以推荐。
- 语言（可选项）：选择语言。
- 文章文字（可选项）：文章的描述文字。

在“新文章”区域填写完文章的基础信息后，需要对文章的权限进行设置。文章权限区域如图 22.48 所示。从图中可以看到，该文章可以分别针对每个前台用户角色进行权限设定，下面以 Public 角色为例进行说明。

基本操作对应的权限说明如下。

- ❑ 删除：删除指定类别权限。
- ❑ 编辑：修改编辑类别权限。
- ❑ 编辑状态：修改编辑状态权限。

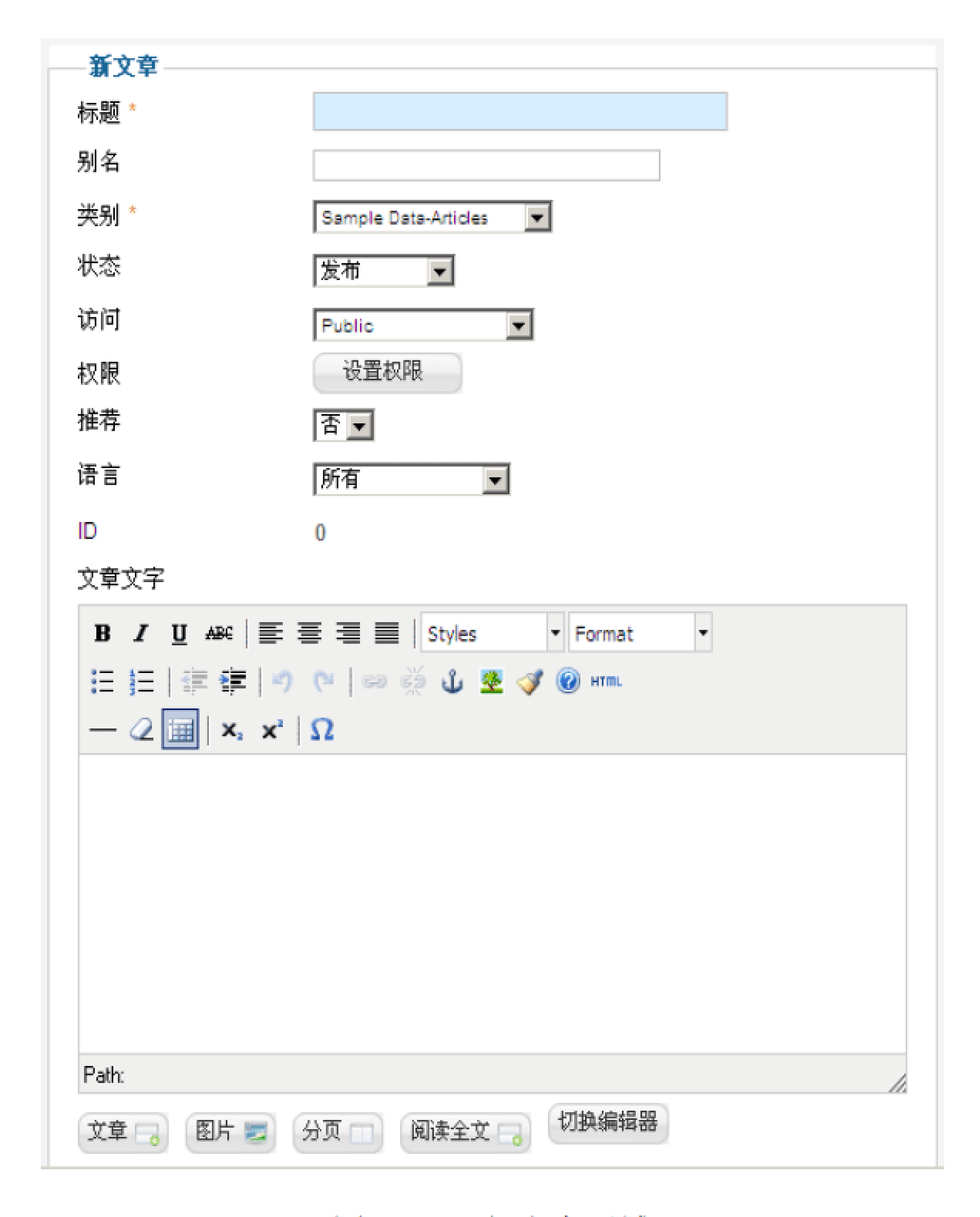


图 22.47　新文章区域



图 22.48　文章权限区域

设置完文章的权限后，选择一个“建立者”选项，单击“保存”命令，完成新建文章操作，如图 22.49 所示。



图 22.49　确认新建文章

2．文章管理功能

前面已经向内容管理系统中录入了文章（内容），下面将介绍对已有内容（文章）的基本管理操作。有很多功能并不复杂，读者可以在实际使用中测试使用。现在介绍几个典型的管理功能。

- 文章编辑功能：对已经录入系统的文章进行编辑操作，包括修改内容和修改文章权限。
- 文章发布/取消发布功能：设置文章的发布和取消发布状态。
- 文章回收站功能：删除文件还原。

文章编辑功能是管理后台中使用频率最高的功能之一。通过该功能可以修改已经存在的文章和修改该文章的权限等操作。

下面来介绍如何操作，首先在后台控制面板中单击“文章管理”图标进入内容管理的文章列表界面。文章管理界面中可以通过 5 种条件获得文章列表，分别是文章状态、文章类别、访问权限、文章作者和文章语言。

这里选择默认条件，勾选需要编辑的文章，然后单击“编辑”选项，进入文章管理中的编辑文章界面，如图 22.50 所示。



图 22.50　编辑文章

文章编辑的选项和新建文章的项目相同，在前面新建文章时已经介绍过各项的含义了，这里不再重复说明。修改完指定的项目后，单击“保存”图标完成该次修改。

文章编辑是文章管理的主要功能，下面介绍文章管理中的其他辅助功能。

由于 Joomla 的文章发布机制中需要设定文章的发布状态，因此文章管理中提供了文章发布/取消发布功能。在文章列表界面选择指定文章单击“发布”或者“取消发布”图标，完成发布/取消发布操作，如图 22.51 所示。



图 22.51　文章发布/取消发布

在实际的维护管理工作中，常会出现误删除文章的情况。因此在文章管理后台中提供了一个比较实用的功能，就是文章回收站功能，下面介绍文章回收站的使用。

首先在文章列表界面选择待删除文章，单击“回收站”选项，完成将文章放入回收站操作。然后，在文章列表的条件选项中的“选择状态”项，选择“丢到回收站”选项可以看到刚刚删除的文章，如图 22.52 所示。

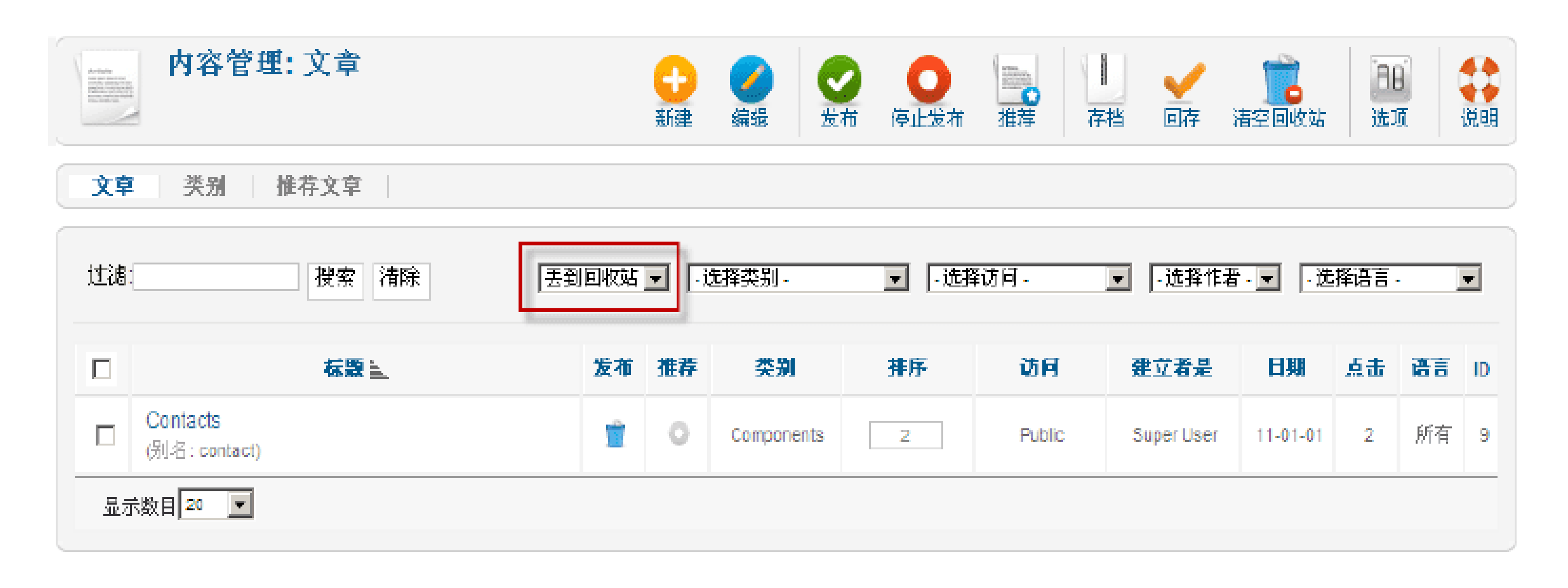


图 22.52　文章回收站列表

当需要恢复文章的时候，在列表页勾选需要恢复的文章，然后单击“回存”选项，完成将回收站中文章恢复的操作。

22.7　插件、组件和模块开发

前面介绍了 Joomla 的基本原理和管理功能，但是如果只是使用系统默认的功能，Joomla 的使用范围将大打折扣。开发者真正应该关注的是，如何基于 Joomla 这个平台开发适合相应需求的扩展应用。前文提到 Joomla 的 3 种标准扩展方式，即插件、组件和模块，本章将详细为读者介绍这 3 种扩展方式的原理，并结合实例开发出简单的应用。

22.7.1　插件详解

Joomla 中引入插件（Plugin）主要是用于预处理，在一些特定事件发生之前或者之后进行的一些处理，比较常见的有登录的授权插件（在登录验证之后进行的授予相应权限的操作）；插件（Plugin）也常被用来设计完成某个独立的功能。在基于 Joomla 平台的开发中，插件（Plugin）的方式是很有效的方式之一。

1．插件的工作原理

对于熟悉内容管理系统（CMS）开发的读者来说插件（Plugin）并不陌生，在很多成熟的 CMS 系统中都有插件扩展。内容管理系统（CMS）中的插件（Plugin）本身是一种事件响应程序。Joomla 中的插件（Plugin）可以响应内核事件和用户自定义事件，可以简单地理解为插件（Plugin）的执行是 hook（勾住）特定的函数（响应事件），然后触发该事件。

Joomla 为了符合面向对象的框架，引入了 JPlugin。JPlugin 本身是一个类程序，该事件处理函数拥有自己的变量和私有的 helper 方法，这种方式使得整个事件处理逻辑非常清晰。这里需要说明的是 Joomla 中的插件（Plugin）是依附于全局事件分派系统的观察者类，也就是说无论是 Joomla 核心，还是第三方组件、模块都能触发一个或多个 JPlugins 来执行一些功能。

JPlugin 系统的是通过观察者模式实现的，它包括两部分，即观察者（JPlugin）和被观察者（JEventDispatcher）。

注意：关于观察者模式，读者可以查阅相关的设计模式手册，这里不做详细说明。

JPlugin 类的代码如下：

```
abstract class JPlugin extends JEvent
{
    public $params = null;
    protected $_name = null;
    protected $_type = null;
    //实例化构造函数
    public function __construct(&$subject, $config = array())
    {
        //获得相关参数
        if (isset($config['params']))
        {
            if ($config['params'] instanceof JRegistry) {
                $this->params = $config['params'];
            } else {
                $this->params = new JRegistry;
                $this->params->loadJSON($config['params']);
            }
        }
        //获得插件名称
        if (isset($config['name'])) {
            $this->_name = $config['name'];
        }
        //获得插件类型
```

```
        if (isset($config['type'])) {
            $this-> type = $config['type'];
        }
        //继承父类构造函数
        parent::  construct($subject);
    }
    /**
    加载插件语言文件
     */
    //加载扩展语言
    public function loadLanguage($extension = '',$basePath = JPATH_ADMINI
    STRATOR)
    {
        if (empty($extension)) {        //检查该扩展是否存在
            $extension = 'plg '.$this-> type.' '.$this-> name;
        }
        //工厂模式获得扩展语言
        $lang = JFactory::getLanguage();
        return
            $lang->load(strtolower($extension),$basePath,null,false, false)
        ||  $lang->load(strtolower($extension), JPATH PLUGINS.DS.$this->
         type.DS.$this-> name, null, false, false)
        ||  $lang->load(strtolower($extension),$basePath, $lang->getDe
        fault(), false, false)
        ||  $lang->load(strtolower($extension), JPATH_PLUGINS .
DS.$this-> type.DS.$this-> name, $lang->getDefault(), false, false);
    }}
```

【代码解读】

JPlugin 类是一个观察者模式的构造器，通过注册观察者并设置观察目标，将 JPlugin 与被观察者建立了联系，所有的 plugin 都观察 JEventDispatcher 对象。

2. 插件类型及目录说明

前面提到 Joomla 的插件可以响应内核事件和用户自定义事件，核心的插件类型主要分为以下几类。

- Authentication：用于不同资源的使用认证，如果需要登录 Joomla 站点，则必须通过 Joomla 数据库里的用户认证。只要任何资源拥有公开的 API，就可以开发认证插件用于认证。
- Content：用于文章内容的输出显示。
- Editors：用于新增一个编辑器，通常是“可见即可得”的编辑器。
- Editors-xtd：编辑器扩展（editors-extend），用于新增编辑器按钮，如图片、视频等按钮。
- Search：当使用搜索组件进行搜索时，将触发相应搜索事件。
- System：主要用于控制 Joomla 在执行 PHP 代码时的行为。
- User：用于相应用户事件的插件，如用户登录等。
- Xmlrpc：提供 Xml-rpc 网络服务时的插件。

打开在 Joomla 的安装目录下的 plugin 文件夹，可看到每种插件类型的相应文件夹，如图 22.53 所示。

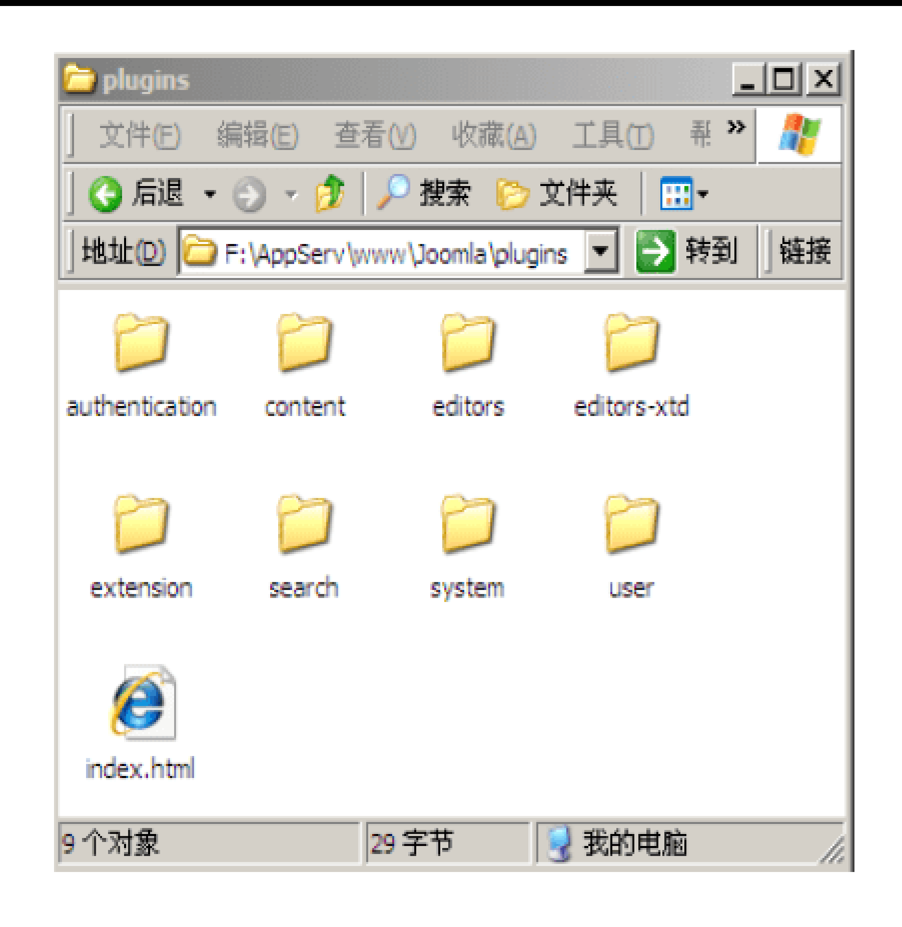


图 22.53　插件类型对应的文件夹

3．插件开发流程

在了解了 Joomla 中插件（Plugin）的基本原理和事件响应类型后，开始制作一个简单的 Joomla 插件。通过创建一个标准的目录搜索机器人，来了解插件（Plugin）开发的基本步骤。

通常标准的 Joomla 插件（Plugin）包含一个单独的代码文件。标准化的安装插件流程要求，必须将插件（Plugin）文件打包成 Joomla 安装程序（installer）能够识别和处理的安装文件。读者在这部分内容里需要掌握插件开发的基本流程，流程步骤如下。

（1）创建安装文件。

（2）创建插件。

（3）在代码中调用插件。

1．创建安装文件

由于目录搜索插件属于搜索插件这一类，因此，首先在根目录的插件目录 plugins 下的搜索插件目录（search）中，创建一个单独的目录（searchbot），如图 22.54 所示。

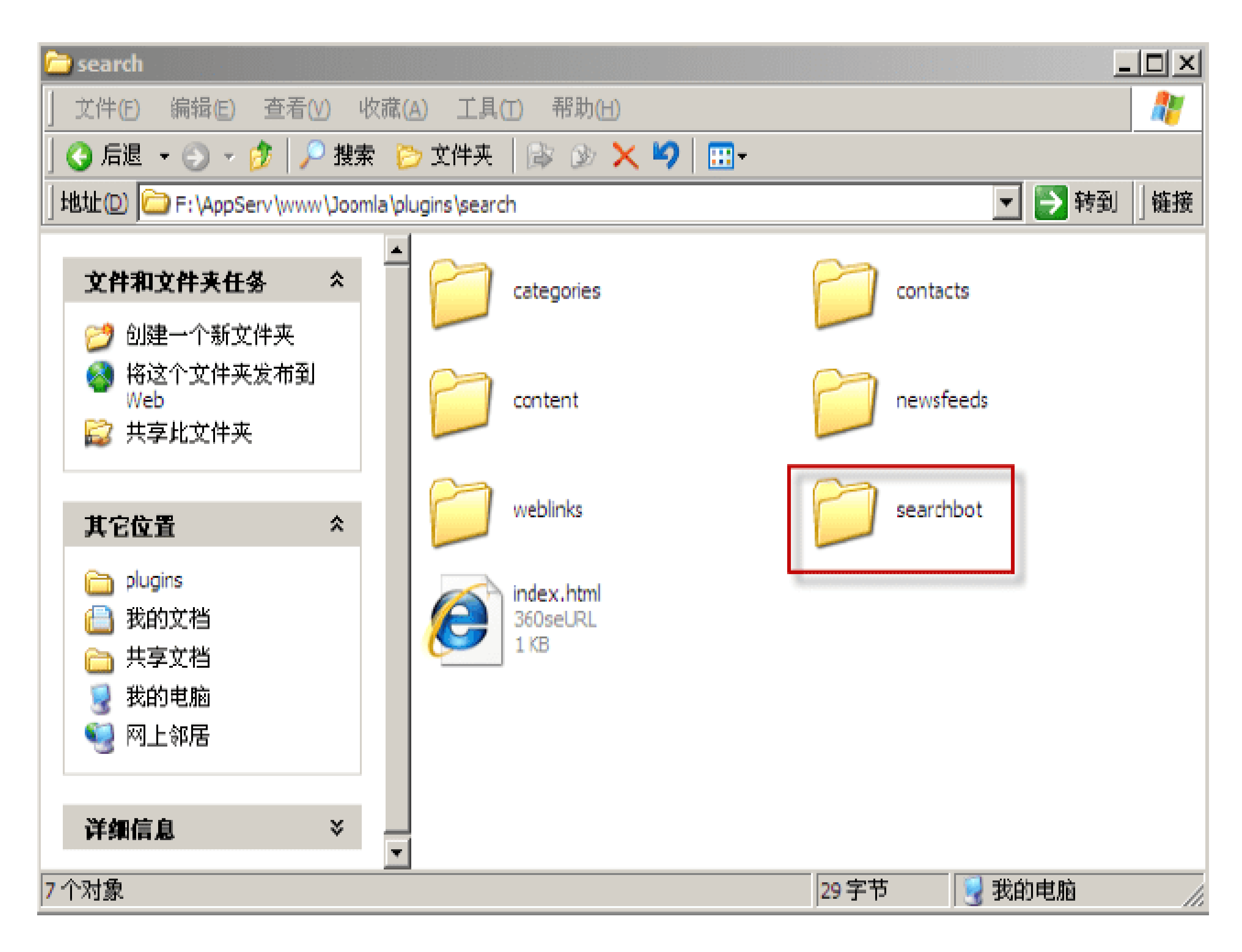


图 22.54　新建插件目录

打开 searchbot 目录创建安装文件（searchbot.xml），该文件是一个标准的 XML 文件，用来记录插件的安装信息。该文件的代码如下：

```
<?xml version="1.0" encoding="iso-8859-1"?>
<install version="1.5" type="plugin" group="search">
    <name>Categories searchbot</name>
    <author>Joomla! Project</author>
    <creationDate>November 2005</creationDate>
    <copyright>(C) 2005 Open Source Matters. All rights reserved.</copyright>
    <license>GNU/GPL</license>
    <authorEmail> admin@joomla.org</authorEmail>
    <authorUrl>www.joomla.org</authorUrl>
    <version>1.1</version>
    <description>Allows searching of Categories information</description>
    <files>
        <filename plugin="categories.searchbot">categories.searchbot.php
        </filename>
    </files>
    <params>
        <paramname="search limit"type="text"size="5"default="50" label=
        "Search Limit" description="Number of search items to return"/>
    </params>
</install>
```

【代码解读】

在<install>标签中的 group 属性（group=“xxx”）表示该插件属于哪个插件类别、插件文件属于哪个文件夹。在<filename>标签中的信息表示插件的文件名，其他标签是一些基础的配置信息。

2．创建插件

下面开始创建插件文件，该文件命名为 searchbot.php。该文件用来完成目录搜索的业务逻辑，代码如下：

```
<?php
//禁止直接访问
defined( ' JEXEC' ) or die( 'Restricted access' );
// 导入所需的库文件
jimport('joomla.event.plugin');
class plg<PluginGroup><PluginName> extends JPlugin
{
    /**
     * 构造函数
     * 为兼容 php4，我们不能直接使用 __constructor 作为插件的构架函数，因为
    func get args ( void )
     * 返回的是所有参数的拷贝，而不是引用，在交叉引用——这在观察者(observer)设计模
    式中是必须的——
     * 将导致错误
     */
    function plg<PluginGroup><PluginName>( &$subject )
    {
        parent::  construct( $subject );
        // load plugin parameters
        $this-> plugin = > JPluginHelper::getPlugin( '<GroupName>',
        '<PluginName>' );
        $this->_params = new JParameter( $this->_plugin->params );
```

```
    }
    /**
     * 与事件同名的插件方法将被自动调用
     */
    function <EventName>()
    {
        global $mainframe;
        return true;
    }
}
? >
```

【代码解读】

上面的插件程序（searchbot.php）很简洁，同样也是一个标准的插件代码样式。这里面用到了前文提到的 Jplugin 类，通过对该类的集成，实现了对多个任意事件的注册，实现了插件的事件调用。

打开刚才创建的目录搜索插件文件夹（searchbot），如图 22.55 所示。该文件夹包含两个目录，即搜索程序（searchbot.php）和插件安装文件（searchbot.xml）。至此，目录搜索机器人插件创建完毕。

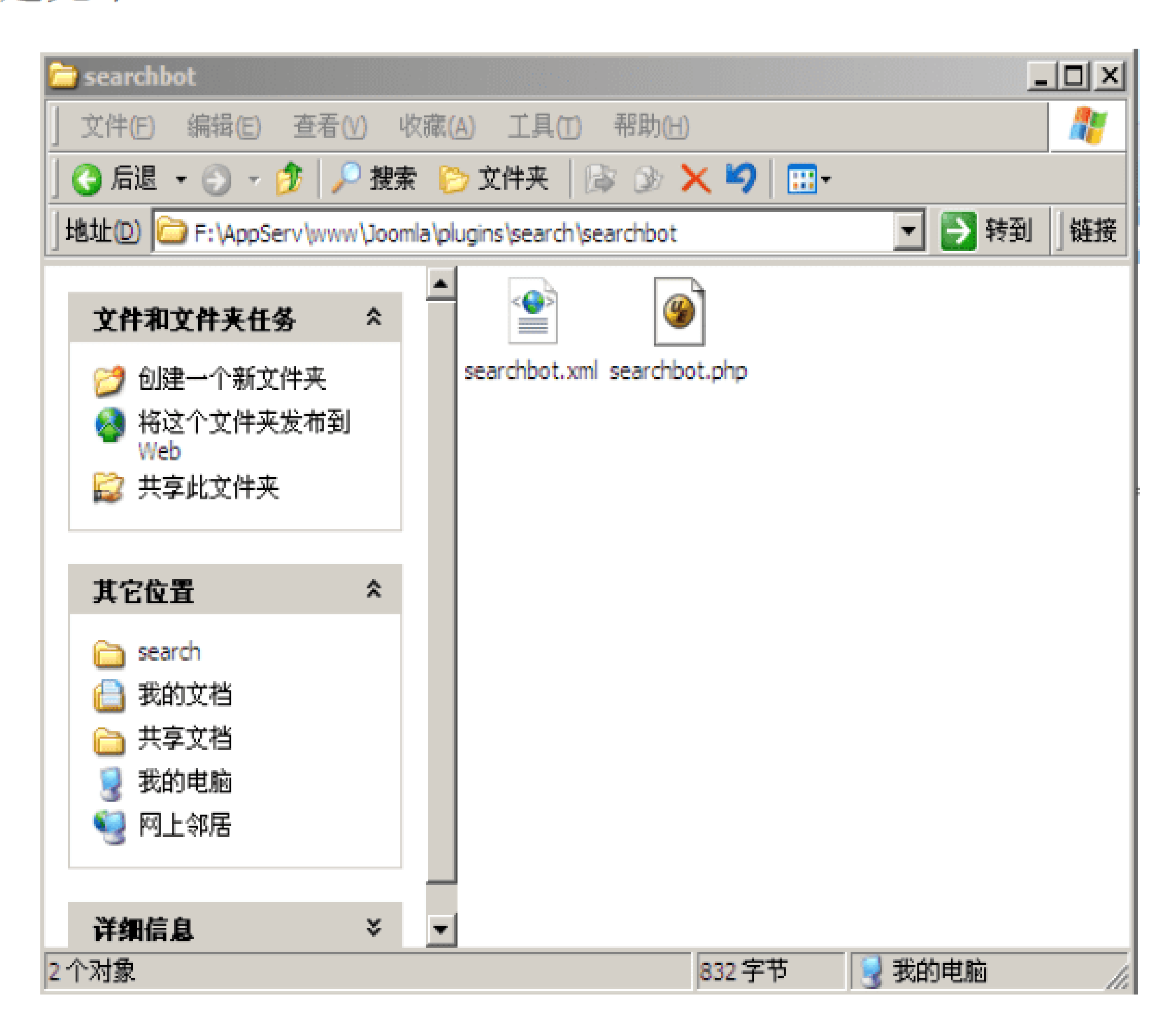


图 22.55　目录搜索文件夹

3. 在代码中调用插件

前面已经创建好了目录搜索机器人插件，Joomla 的插件调用规定，可以使用内核中包括了内置事件和用户自定义的事件调用插件，下面通过一个简便的方法来实现对该插件的调用。触发该插件事件的编码如下：

```
$results = $mainframe->triggerEvent( '<EventName>', <ParameterArray> );
```

【代码解读】

该调用事件包含两个标签，标签的说明如下。

- EventName：事件或函数的名称。
- ParameterArray：事件参数，数组形式的参数。如果是多个参数，需要写在一个数组中，插件函数会遍历这些参数。

22.7.2　组件开发

组件的方式是 Joomla 扩展的最主要形式，Joomla 本身就是基于组件的设计模式。下面将为读者介绍组件的基础知识、MVC 的组件开发模式、Joomla 中组件的标准结构规划，然后完成 Hello World 组件的开发。

1. 组件开发基础

Joomla 组件开发的过程是使用框架（MVC）开发的过程，Joomla 后台也提供了统一的组件管理功能，使用者可以通过该后台对组件进行维护。在开始正式的组件开发之前先介绍组件开发的一些基础知识。内容要点如下。

- Joomla 组件的架构模式（MVC）：Model-View-Controller（MVC）是一种软件设计模式。它被用来组织业务逻辑和数据存储分离的程序代码，Joomla 的组件就是基于这个架构模式。
- 目录文件结构规划：标准组件目录设计及规划。
- 组件数据库概述：组件相关的数据库说明。

Model-View-Controller（MVC）模式被用来组织 Joomla 组件的架构。熟悉 Web 开发的读者对 MVC 的概念应该不陌生，这里简单地回顾一下 MVC 的基本概念，然后介绍 Joomla 中的 MVC 的实现。

按照官方文档中的描述，MVC 各部分的含义如下。

- 模型（Mode）：模型是组件中封装应用程序数据的部分。除了从模型中获取数据外，它通常还提供有效的方法来管理和操作数据。一般来说，底层的数据访问技术应该被封装在模型中。这样，如果将应用程序从顺序文件存储系统迁移到数据库系统，则仅仅需要改变模型单元，而不是视图或控制器。
- 视图（View）：用户交互模型中以恰当的方式向用户呈现数据的部分。对于基于 Web 的应用程序，视图通常指的是返回数据的 HTML 页面。视图从模型获取数据（通过控制器传递给视图），将数据注入到合适模板，并呈现给用户。视图不会引起任何数据变更，它仅显示从模型中获取的数据。
- 控制器（Controller）：控制器负责响应用户的动作。在 Web 应用程序的案例中，用户动作通常是一个页面请求。控制器将决定用户建立哪些请求，通过激活合适的数据操作模型对请求做出恰当的响应，并将模型传递给视图。控制器不直接在模型中显示数据，它仅仅触发修改数据的模型方法，然后将模型传递给显示数据的视图。

在 Joomla 中，MVC 模式的实现主要依靠 3 个类来完成，即 JModel 类、JView 类和 JController 类。

（1）JModel 类是负责 MVC 架构中的数据视图部分。需要注意的是，Joomla 中 JMode 不等同于 JTable 类，数据视图是由一个或多个 JTable 构成的，或者是一个数据集合及数据集合的相关操作。在组件开发过程中，通常都要继承 JModel 类，在子类中完成数据集合及相关的保存和删除操作。

（2）JController 类是 MVC 的控制器部分，通常控制器是程序的入口，它决定下面的

动作。在 JController 类根据提交参数（task）调用 JController 中的相应方法实现相应的业务逻辑。

（3）JView 类中页面的 HTML 代码是由 JView 和 tmpl 目录中的模板共同决定的。JView 可以看作是 JModel 和 Template 之间的“桥梁”，扩展组件的开发通常都需要对 JView 的子类进行扩展。

下面以 Joomla 中的媒体组件为例，说明 Joomla 标准组件的目录文件结构。通过这个例子读者可以对组件的目录规划有清晰的了解。媒体组件的目录结构如图 22.56 所示，通过该图可以看到目录中包含 5 个文件和 4 个文件夹，含义如下。

- Administrator 文件夹：管理后台根目录文件。
- Compontent 文件夹：组件根目录。
- com_media 文件夹：媒体组件目录。
- media.php：组件主程序文件。
- controllers 文件夹：组件控制器文件夹。
- helpers 文件夹：组件帮助文件。
- models 文件夹：组件业务模型文件夹。
- views 文件夹：组件视图文件夹。
- controller.php：组件控制器文件。
- config.xml：组件配置文件。
- media.xml：组件安装文件。

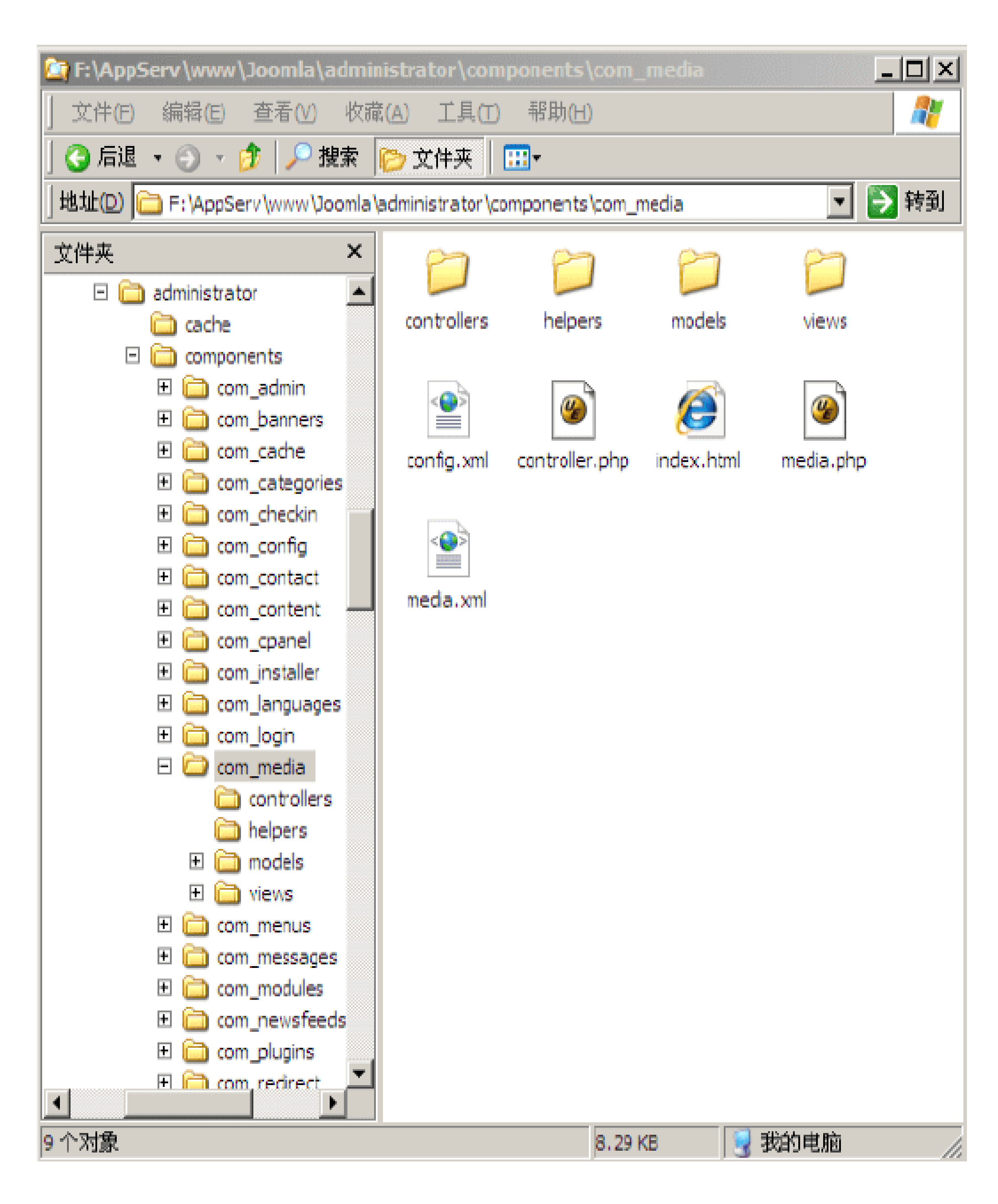


图 22.56　媒体组件目录结构

在 Joomla 安装成功后，默认的设置会将所有的组件数据写入组件数据表中。该表名为 jos_components，该组件数据表中字段的描述如表 22.16 所示，读者参照该表加深对组件开发的理解。

表 22.16　组件数据表字段含义

字　段	类　型	含　义
id	int(11)	自增 id
name	varchar(150)	组件名
link	varchar(255)	组件链接
menuid	int(11) unsigned	菜单项目外键
parent	int(11) unsigned	组件继承关系
admin_menu_link	varchar(255)	对应后台管理菜单功能项
admin_menu_alt	text	后台管理菜单功能项描述
option	varchar(50)	操作类型
ordering	int(11)	排序
admin_menu_img	varchar(255)	后台管理菜单功能项图片
params	text	组件参数
enabled	tinyint(4)	标识位（启用/禁用）

2．建立组件

Hello World 组件是一个基本组件，功能非常简单。在组件目录文件夹下面建立 com_hello 文件夹，用来保存 Hello World 组件的相关文件。该文件夹下面包含 5 个文件，如图 22.57 所示，文件功能如下。

- hello.php：该文件是组件的入口文件。
- controller.php：该文件是组件的主控制器文件。
- /views/hello/view.html.php：该文件获取必要的数据同时写入数据到模板。
- /views/hello/tmpl/default.php：该文件是输出模板文件。
- hello.xml：该文件是用于 Joomla 安装的 XML（manifest）配置文件。

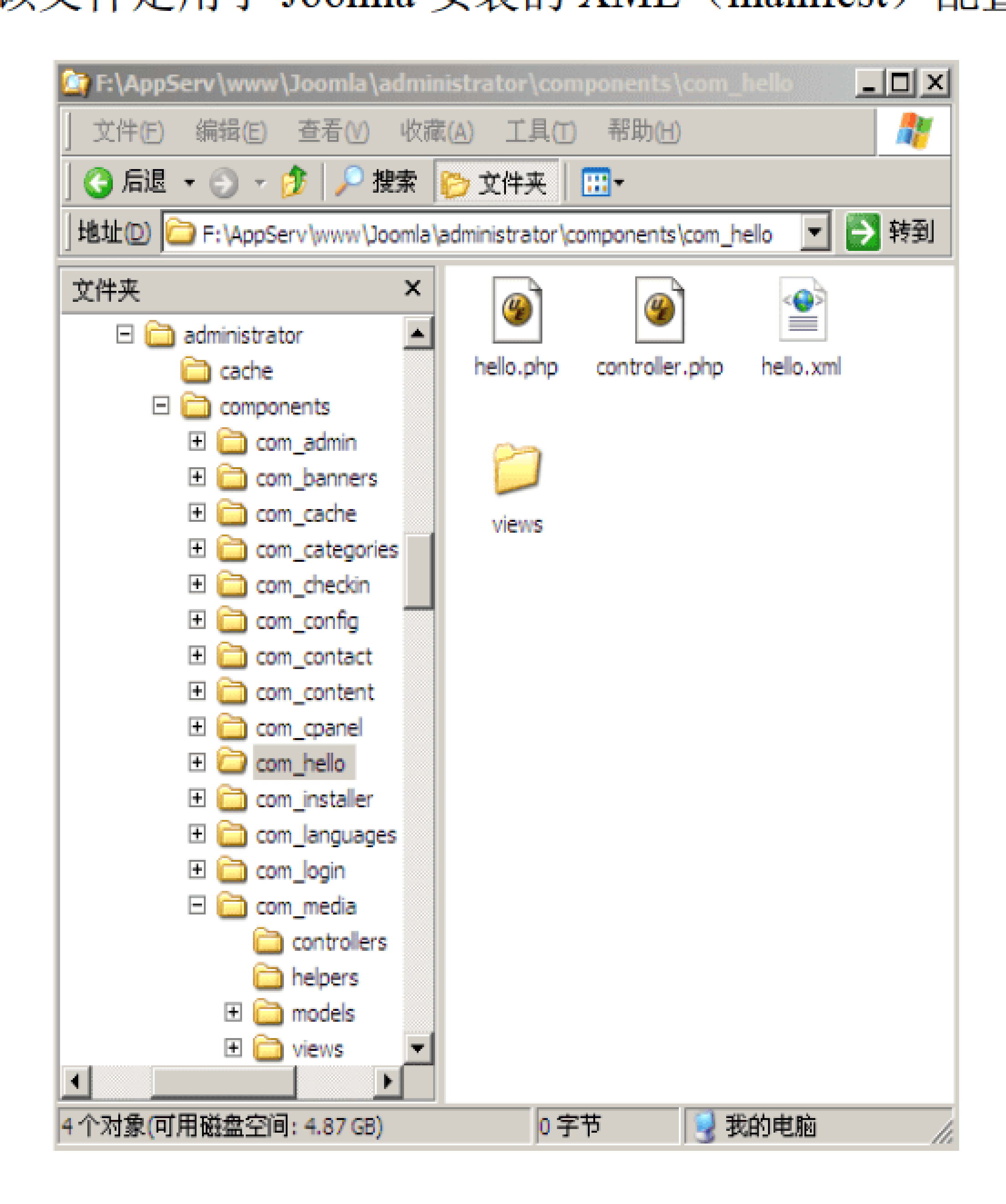


图 22.57　Hello World 组件目录

3．组件入口文件

Joomla 组件是通过单一的入口来访问，前台网站应使用 index.php 文件来访问，后台管理员使用 administrator/index.php 来访问，组件程序根据 URL 或 POST 数据中的 option 的值来加载请求的组件。调用的 HelloWorld 组件 URL 如下：

```
index.php?option=com_hello&view=hello
```

【代码解读】

当程序入口文件调用该链接时，Joomla 会调用/components/com_hello/hello.php 文件。该文件就是 Hello World 组件的入口文件，该文件代码如下：

```
<?php
/**
 *Hello World模型组件
 */
//访问控制（不能被直接调用）
defined( ' JEXEC' ) or die( 'Restricted access' );
//加载控制器基类
require once( JPATH COMPONENT.DS.'controller.php' );
// 控制器响应
if($controller = JRequest::getWord('controller')) {
    $path = JPATH COMPONENT.DS.'controllers'.DS.$controller.'.php';
    if (file exists($path)) {
        require once $path;
    } else {
        $controller = '';
    }
}

//创建控制器
$classname    = 'HelloController'.$controller;
$controller   = new $classname( );
//执行任务请求
$controller->execute( JRequest::getWord( 'task' ) );
//检测跳转指向
$controller->redirect();
```

【代码解读】

以上程序执行步骤如下：

（1）检查是否文件被直接引用。

（2）加载控制器基类。

（3）使用 JRequest:getWord()获取 URL 或 POST 数据中的字符变量。

（4）获取组件所使用控制器的名字。

（5）创建控制器。

（6）创建控制器后，如果程序执行请求中有参数则执行参数请求，默认将加载 View 视图文件或输出结果。

（7）检测跳转指向。

注意：在该文件中使用 PHP 关闭标记“?>”。这样做的原因是，在输出代码中不会有任何不必要的空格。这是 Joomla 用于所有 PHP 文件的默认做法。

4．组件控制器

组件中的控制器用来加载完成业务逻辑模型文件和对应的视图，由于 Hello World 组件没有复杂的业务逻辑，因此该组件控制器只需要显示调用 View 视图文件完成加载即可。Hello World 组件控制器（controller.php）的代码如下：

```
<?php
/**
 *Hello World组件控制器
 */
// 指向检查
defined( ' JEXEC' ) or die( 'Restricted access' );
jimport('joomla.application.component.controller');
/**
 * Hello World组件控制器类
 */
classHelloController extends JController
{
    /**
     * 加载模型和对应的视图
     */
    function display()
    {
        parent::display();
    }
}
```

【代码解读】

由于 Hello World 组件仅有一个逻辑需要处理，即加载视图文件显示 Hello World。因此，控制器非常简单，没有数据操作请求。组件控制器需要做的就是加载指定的视图，控制器调用视图显示方法 JController::display()即可。

5．组件视图

视图的任务非常简单，获取用来显示的数据，并且将其注入到模板视图文件 view.html.php。该文件的代码如下所示：

```
<?php
/**
 *Hello World组件视图类文件
 */
//指向检查
defined( ' JEXEC' ) or die( 'Restricted access' );
jimport( 'joomla.application.component.view');
/**
 * Hello World组件视图类
 */
classHelloViewHello extends JView
{
    functiondisplay($tpl = null)
    {
        $greeting = "HelloWorld!";
        $this->assignRef( 'greeting', $greeting );
```

```
        parent::display($tpl);//输出模板
    }
}
```

【代码解读】

上述代码的核心是使用 JView::assignRef()方法将数据注入到模板。

注意：传递给 assignRef 方法的键值（第一个参数）不能以下划线开头，即$this->assignRef('_greeting',$greeting)。这样做 assignRef 方法会返回 false，并无法将变量值注入到模板中。

6．组件模板

Hello World 组件模板非常简单，作用是显示从视图传递来的 greeting 变量值。该文件（default.php）的代码如下：

```
<?php
/*
 *Hello World 组件模板文件
 */
//指向检查
defined('_JEXEC') or die('Restrictedaccess'); ?>
//模板变量输出
<h1><?php echo $this->greeting; ?></h1>
```

【代码解读】

组件模板文件使用 HTML 标签来显示程序输出的数据，具体是使用 JView::assignRef()方法绑定的变量，在模板中能够通过$this->{propertyname}的方式来访问。

7．组件安装文件

Joomla 中组件的简便的方式是通过修改数据表 jos_components 添加一条记录，标准安装方式是通过 XML 安装文件的方式进行安装。Hello World 组件的 XML 安装文件代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<install type="component" version="1.5.0">
<name>Hello</name>
<creationDate>2007-02-22</creationDate>
<author>JohnDoe</author>
<authorEmail>joomla@example.org</authorEmail>
<authorUrl>http://www.example.org</authorUrl>
<copyright>CopyrightInfo</copyright>
<license>LicenseInfo</license>
<version>1.01</version>
<description>Descriptionof the component ...</description>
<files folder="site">
<filename>controller.php</filename>
<filename>hello.php</filename>
<filename>index.html</filename>
<filename>views/index.html</filename>
<filename>views/hello/index.html</filename>
<filename>views/hello/view.html.php</filename>
<filename>views/hello/tmpl/default.php</filename>
```

```
<filename>views/hello/tmpl/index.html</filename>
</files>

<administration>
<menu>HelloWorld!</menu>
<files folder="admin">
<filename>hello.php</filename>
<filename>index.html</filename>
</files>
</administration>
</install>
```

【代码解读】

上述代码是一个标准的 Joomla 安装 XML 文件，该文件定义的内容如下：

- 组件的基本描述信息（如名字等）、选项、说明、版权和许可信息。
- 需要复制文件的清单。
- 可选，执行额外安装和卸载操作的 PHP 文件。
- 可选，在安装/卸载过程执行数据库查询的 SQL 语句。

22.7.3　模块开发

在 Joomla 中，模块是一种简便的扩展的方式，相对组件更为灵活。模块通常用来做页面中不太复杂的一小块，并且能够跨越不同的组件。很多时候模块与组件密切联系，在 Joomla 中常见的模块应用的例子很多（如最新新闻模块、登录框模块）。Joomla 中的一些核心的应用也是由模块构成（如菜单模块），这些模块作为 Joomla 和核心应用，并没有和任何组件相关联。

1．模块开发基础

Joomla 中的模块具有很强的独立性，它不需要和任何特定的组件相关联，所以通常的应用就是通过模块来扩展一个独立的页面或者单独显示区域。下面的例子中也将通过一个 Hello World 模块的例子介绍模块开发的基本步骤，该模块将在网站前台上显示一个空白的页面。

首先打开 Joomla 根目录下的 modules 文件夹，该文件夹用来存放对应的模块程序，命名规则为“mod_***”其中“***”为模块名称。模块文件目录结构如图 22.58 所示。

标准的模块应用包含 4 个基本文件，下面以即将要开发 Hello World 模块为例说明每个文件的含义。

- mod_helloworld.php：模块的入口文件，该文件主要执行一些必须的初始化工作，调用 helper 方法获得必要的数据，并引入模板。
- mod_helloworld.xml：这个文件主要包含模块的信息，主要定义安装时必须的文件以及模块的参数。
- helper.php：这个文件包含 helper 类，该类主要用来获取模块要显示的信息（通常是从数据库或其他的源）。
- tmpl/default.php：模块的模板，该文件采用 mod_helloworld.php 返回的数据生成页面要显示的 HTML 页面。

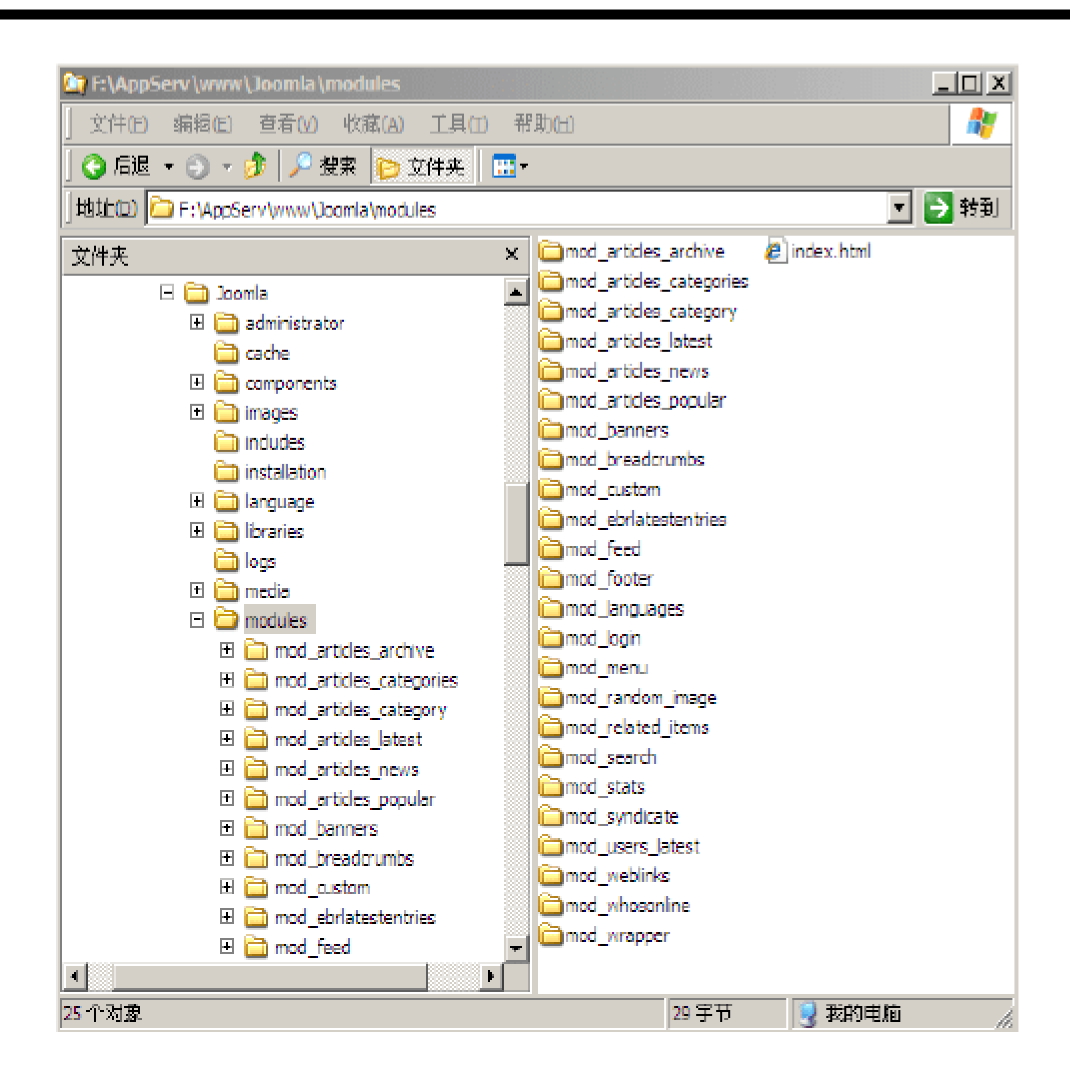


图 22.58　模块文件夹目录结构

打开 Joomla 根目录下的 modules 文件夹，然后创建一个名为 mod_hello 的文件夹，该文件夹包含 Hello World 模块程序。进入该文件夹目录结构如图 22.59 所示。

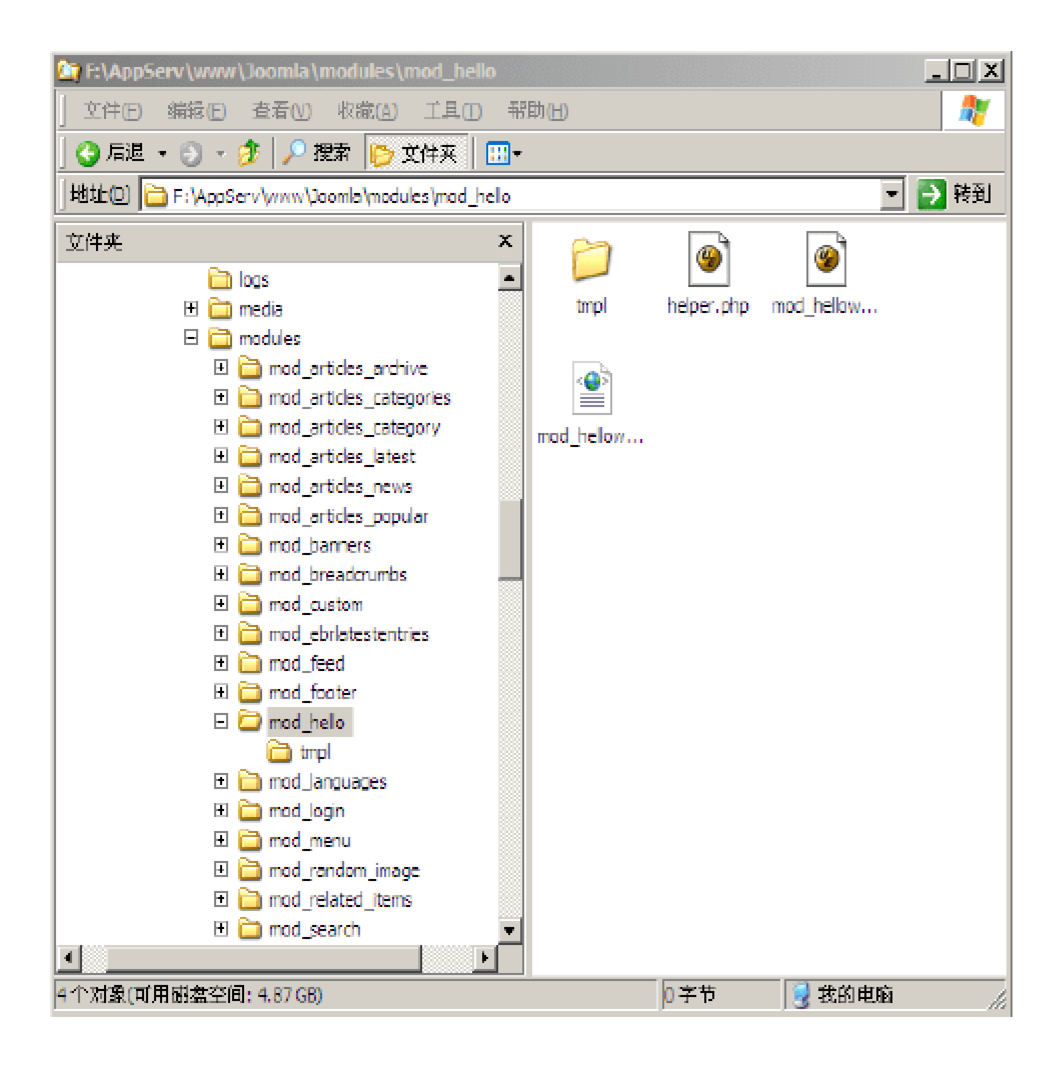


图 22.59　Hello World 模块文件目录结构

2．模块数据库说明

在 Joomla 安装成功后，默认的设置会将所有的模块数据写入模块数据表中，该表名为 jos_modules。该组件数据表中字段的描述如表 22.17 所示，读者参照该表了解模块中各字段的含义。

表 22.17　模块数据表

字　段	类　型	说　明
id	int(11)	自增 id
title	text	模块名称
content	text	模块内容描述

续表

字　段	类　型	说　明
ordering	int(11)	显示排序
position	varchar(150)	显示位置
checked_out	int(11) unsigned	
checked_out_time	datetime	
published	tinyint(1)	是否发布
module	varchar(150)	使用模块名称
numnews	int(11)	
access	tinyint(3) unsigned	
showtitle	tinyint(3) unsigned	是否显示名称
params	text	参数
iscore	tinyint(4)	
client_id	tinyint(4)	客户 id
control	text	

3. 模块开发

前面介绍了模块开发的基础和数据库描述，模块开发的基本步骤如下。

（1）创建入口文件

下面逐步为读者详细地介绍每个组成步骤。首先创建模块的入口文件（mod_helloworld.php），该文件的代码如下：

```
<?php
/**
* Hello World 模块入口程序
*/
//实例化应用，防止注入攻击
defined( '_JEXEC' ) or die( 'Restricted access' );
// 引入 helper.php 类程序
require_once( dirname(__FILE__).DS.'helper.php' );
$hello = modHelloWorldHelper::getHello( $params );
require( JModuleHelper::getLayoutPath( 'mod_helloworld' ) );
```

【代码解读】

上面入口文件的主要工作是调用合适的 helper 类，并返回数据。由于该 Hello World 模块没有用到参数设置，但是为了方便以后扩展模块，需要将参数传递给 helper 的方法。

helper 类的方法用以下的方式调用：

```
$hello = modHelloWorldHelper::getHello( $params );
```

（2）创建 Helper 文件

创建 helper 文件 helper.php 这个文件包含 helper 类，这个类主要用来获取模块要显示的数据。以下是 helper.php 的代码：

```
<?php
/**
 * Hello World 模型中的 Helper 类
 */
class modHelloWorldHelper
```

```
{
    /**
     * 返回 Hello, World!显示信息
     */
    function getHello( $params )
    {
        return 'Hello, World!';
    }
}
```

【代码解读】

以上代码的 helper 类只有一个 getHello()方法，这个方法的作用是返回“Hello World”字符串。

（3）创建模板文件

模块中的模板的作用是 HTML 页面的展示页。模板文件（default.php）的代码如下：

```
<?php
//实例化应用，防止注入攻击
defined( '_JEXEC' ) or die( 'Restricted access' ); ?>
<?php
echo $hello; //输出测试信息
?>
```

【代码解读】

以上代码十分简单，只是输出变量信息。但是需要注意的是，模板文件和 mod_helloworld.php 文件是同一个变量作用域。也就是说，在 mod_helloworld.php 中定义的变量$hello 可以直接在模板文件中使用，而不必额外声明或者函数调用。

（4）创建安装配置文件

安装配置文件（mod_helloworld.xml）主要是用来指定那些文件安装时需要复制的模块文件，以及设置一些模块安装的参数。安装配置文件（mod_helloworld.xml）代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.6.0">
    <name>Hello, World!</name>
    <author>Admin</author>
    <version>1.6.0</version>
    <description>A simple Hello, World! module.</description>
    <files>
        <filename module="mod hello">mod helloworld.php</filename>
        <filename>index.html</filename>
        <filename>helper.php</filename>
        <filename>tmpl/default.php</filename>
        <filename>tmpl/index.html</filename>
    </files>
    <params>
    </params>
</install
```

【代码解读】

上面的安装配置文件（mod_helloworld.xml）中定义了基本的安装信息，其中<files></files>标签中定义模块文件的位置。

（5）模块安装测试

Hello World 模块安装成功后，进入后台模块管理频道，可以看到 Hello World 模块列

表，如图 22.60 所示。

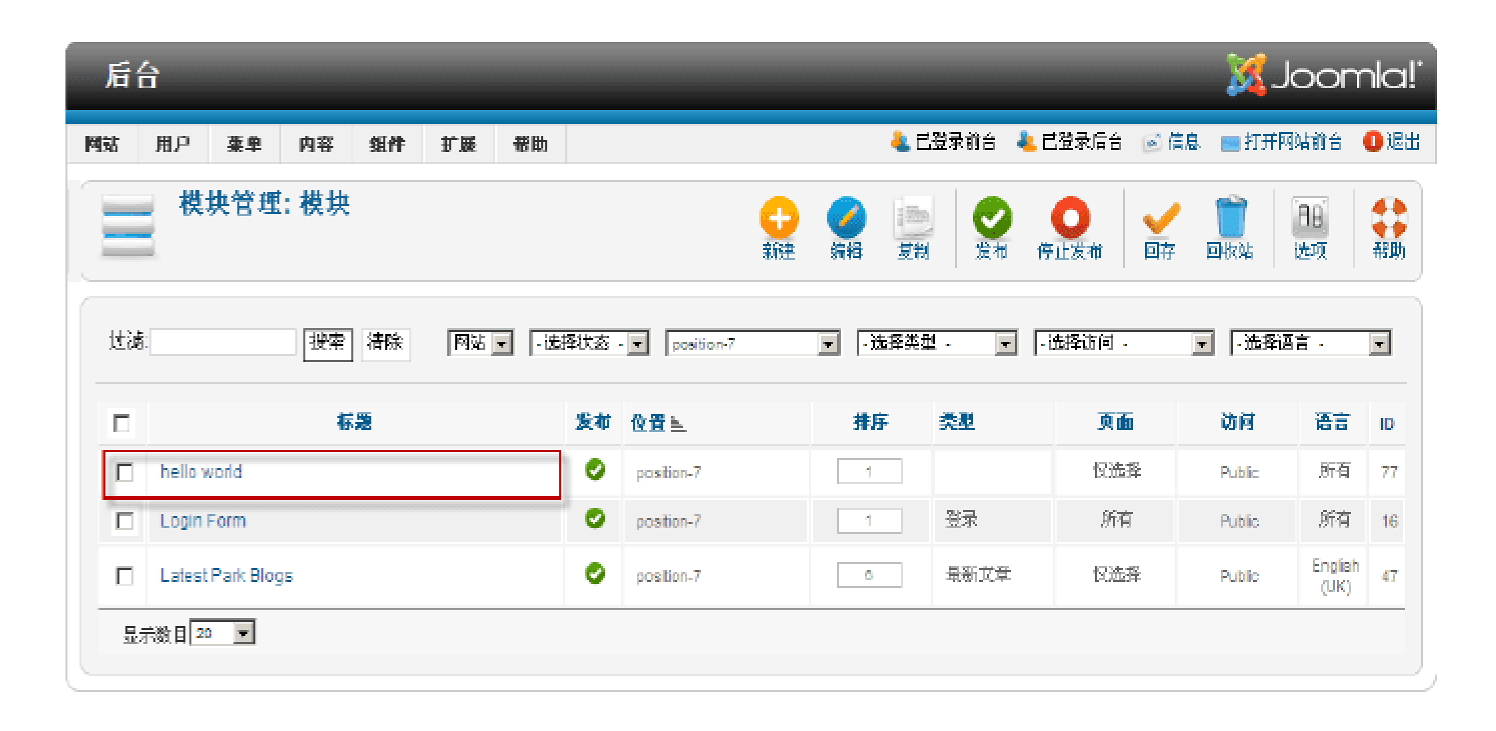


图 22.60　Hello World 模块管理列表

进入 Hello World 模块管理，由于该模块完成的功能比较单一，因此没有配置额外的参数。管理界面如图 22.61 所示。

配置过程中最重要的步骤就是选择模块显示的位置，这里选择“position-7”，然后单击“保存并关闭”按钮完成设置。打开浏览器，在页面中测试模块内容输出，如图 22.62 所示。从图中可以看到，该模块完成了预定信息输出。





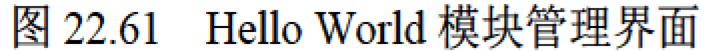


图 22.61　Hello World 模块管理界面　　　　图 22.62　Hello World 模块前台展示

22.8　快速建站实例

Joomla 设计的其中一个原则就是便捷性体现在开发者可以快速开发扩展，同时便于使用者快速搭建应用。本节将通过 Joomla 平台快速搭建一个小型网站，来为读者直观地展示 Joomla 搭建网站的标准流程，同时回顾前几节介绍过的内容。

本节的主要任务如下：

- ❑ 需求分析与设计；
- ❑ 添加网站内容；

- 安装图片展示插件；
- 安装评论组件；
- 安装留言板模块；
- 模板安装及规划；
- 创建导航栏及关联菜单；
- 前台页面测试。

22.8.1　系统介绍

本节将搭建一个小型的企业官方网站，并满足常见的企业用户对外宣传的基本需求。本节中的示例将以最新版本的 Joomla 软件实例化一个企业管理后台，然后通过该后台完成网站的基本设置和内容添加。该网站需要完成的功能如下。

- 用户系统：用户的基本管理，注册用户权限划分。
- 文章发布系统：发布网站内容，并具备基本的管理功能。
- 用户评论系统：实现用户对指定文章和内容的评论（评论组件实现）。
- 产品图文展示功能：展示指定分类或者文件夹的图片（图片展示插件）。
- 留言板功能：提供访客留言的功能，管理者可以通过后台实现基本的管理功能（留言板模块）。

22.8.2　添加网站内容

网站内容的添加是通过 Joomla 的管理后台来实现的，其中最核心的功能是文章管理功能，绝大部分的网站内容都是通过文章添加的方式录入到系统中的。需要提醒读者注意的是，这里提到的“文章”是广义上的文章，不单指一篇文章、一段文字，包括一个 HTML 的页面也是一篇“文章”，“文章”是网站内容的主要展示形式。

下面通过向系统中录入两篇不同文章，来加深对文章的理解，一篇介绍文章和一篇产品图片的文章。

（1）打开管理后台界面，在控制面板界面单击“新建”文章图标，录入一篇标题为“Joomla 介绍”的文章，如图 22.63 所示。

（2）录入相应的文章内容，这里需要注意类别的选择。这里的类别代表该文章的“性质”，这里选择“Sample Data-Articles”类别，该类别表示该文章是一个简单数据的文章。录入文章后需要设置“发布选项”，该区域分为 3 个部分，该区域需要设置一些文章和发布的参数，如图 22.64 所示。

（3）由于该文章是开放给所有用户浏览的，因此将“文章权限”设置为“Public”，该选项表示该文章权限开放给所有用户浏览。

（4）打开管理后台界面，在控制面板界面单击“新建”文章图标，录入一篇标题为“产品图片”的文章，如图 22.65 所示。设置方法步骤和上面介绍的相同，但是文字介绍部分为空，因为该文章是用来作图片展示的“文章”，读者可以理解为是一个页面显示的“容

器”。具体产品图片的显示会在图片展示插件部分做详细的介绍。



图 22.63　新建 Joomla 介绍文章

发布选项	
建立者	admin　选择用户
建立者别名	
建立日期	2011-03-23 11:22:58
开始发布	2011-03-23 11:22:58
发布结束	0000-00-00 00:00:00
修改者	admin
修改日期	2011-03-23 12:46:51
修订	5
点击	21
文章选项	
Metadata选项	

图 22.64　文章发布选项

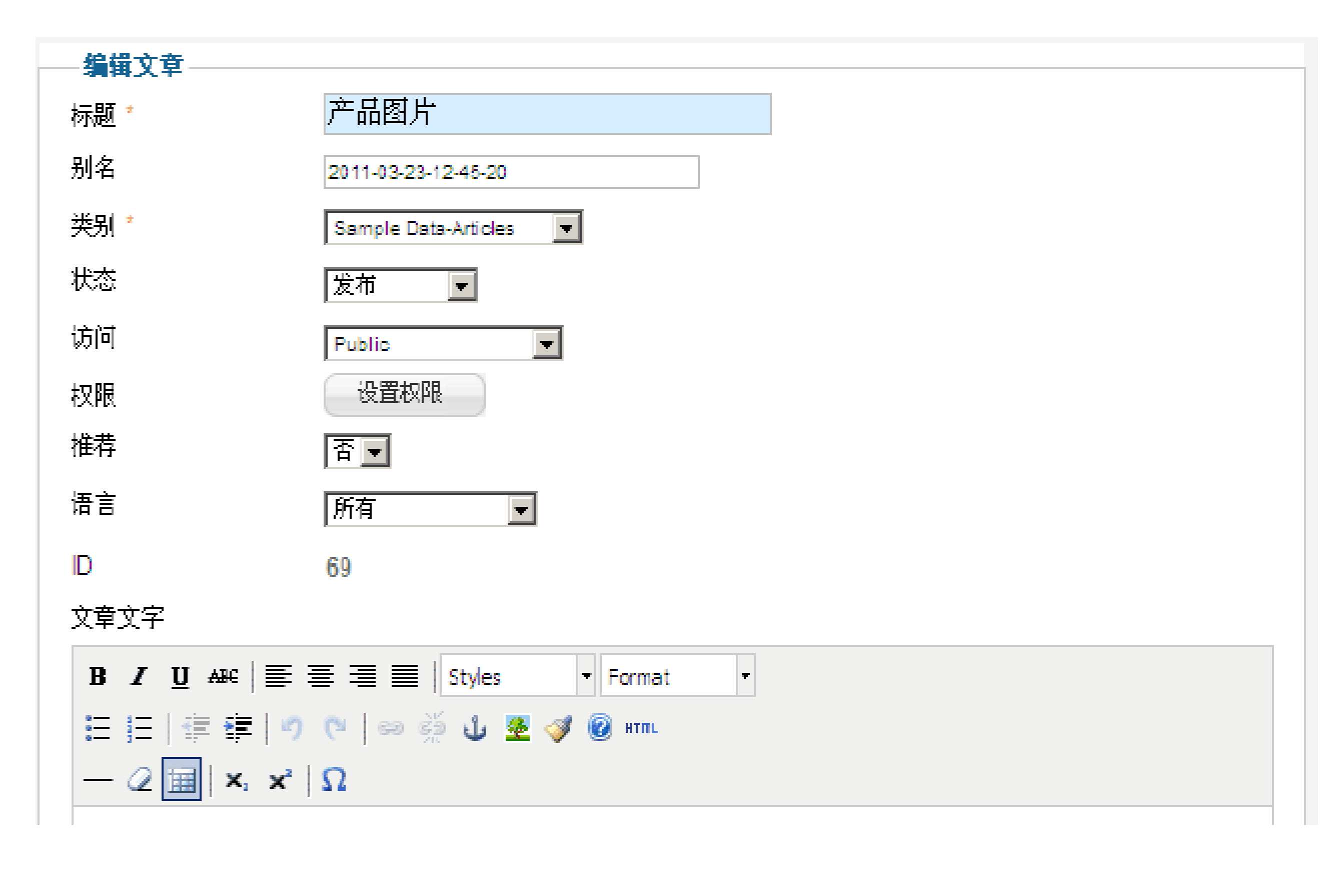


图 22.65　产品图片文章

22.8.3　安装图片展示插件

前面创建了“产品图片”文章，该文章可以被看作是产品图片的页面框架。下面将通过管理后台安装一个开源的图片展示插件，通过将该插件嵌入到产品图片文章中，最终完成产品图片的展示功能。

（1）从 Joomla 官方网站（http://extensions.joomla.org/extensions/）下载名为 Simple Image Gallery 的图片展示插件，如图 22.66 所示。该扩展插件有 1.5 和 1.6 两个版本，读者下载的时候注意选择对应的版本，这里选择该软件的 1.6 版本插件。



图 22.66　图片展示插件下载界面

（2）将该插件下载后保存到本地，然后打开后台管理界面的控制面板，单击“扩展管理”图标，在“安装”选项下选择图片展示插件，单击“上传并安装”按钮，如图 22.67 所示。从图可以看到，该插件已经被正确地安装到系统中。



图 22.67　安装图片展示插件

（3）安装完图片展示插件后，打开文章管理界面勾选“产品图片”文章项目，并单击“编辑”图标。在文章的内容部分插入以下代码，完成对该插件的调用，如图 22.68 所示。

```
{vsig}./phocagallery/{/vsig}
```

【代码解读】

上面是图片展示插件的调用代码，其中{vsig}{/vsig}标签是该插件的调用标记，当程序“读”到该标签后将执行图片展示逻辑代码。./phocagallery/为目录地址，表示将显示该目录下的所有图片。

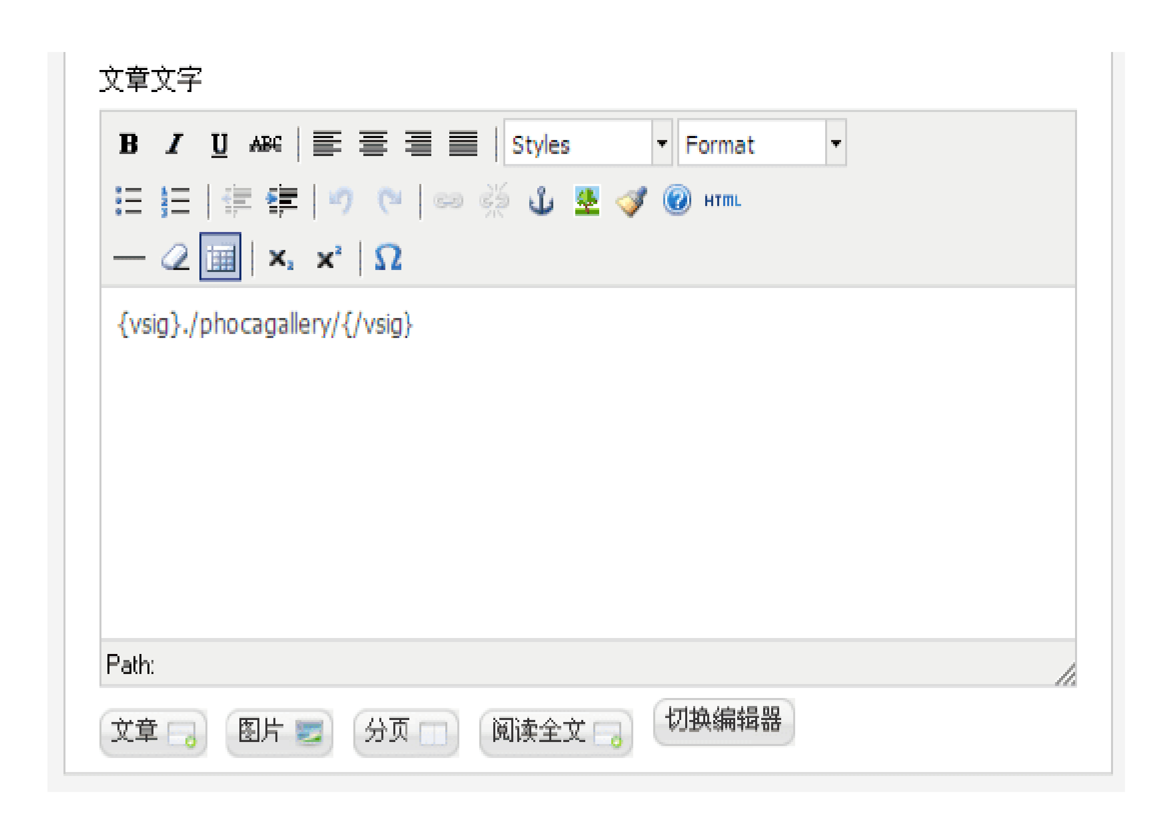


图 22.68　图片展示插件调用

（4）单击“保存”图标完成图片展示文章的修改，预览该文章如图 22.69 所示。该文章完成了产品图片组的展示功能。



图 22.69　产品展示预览

22.8.4　安装留言板组件

首先从 Joomla 官方网站（http://extensions.joomla.org/extensions/）下载名为 easybook reloaded 的留言板组件，如图 22.70 所示。该组件有 1.5 和 1.6 两个版本，读者下载的时候注意选择对应的版本，这里选择该软件的 1.6 版本插件。

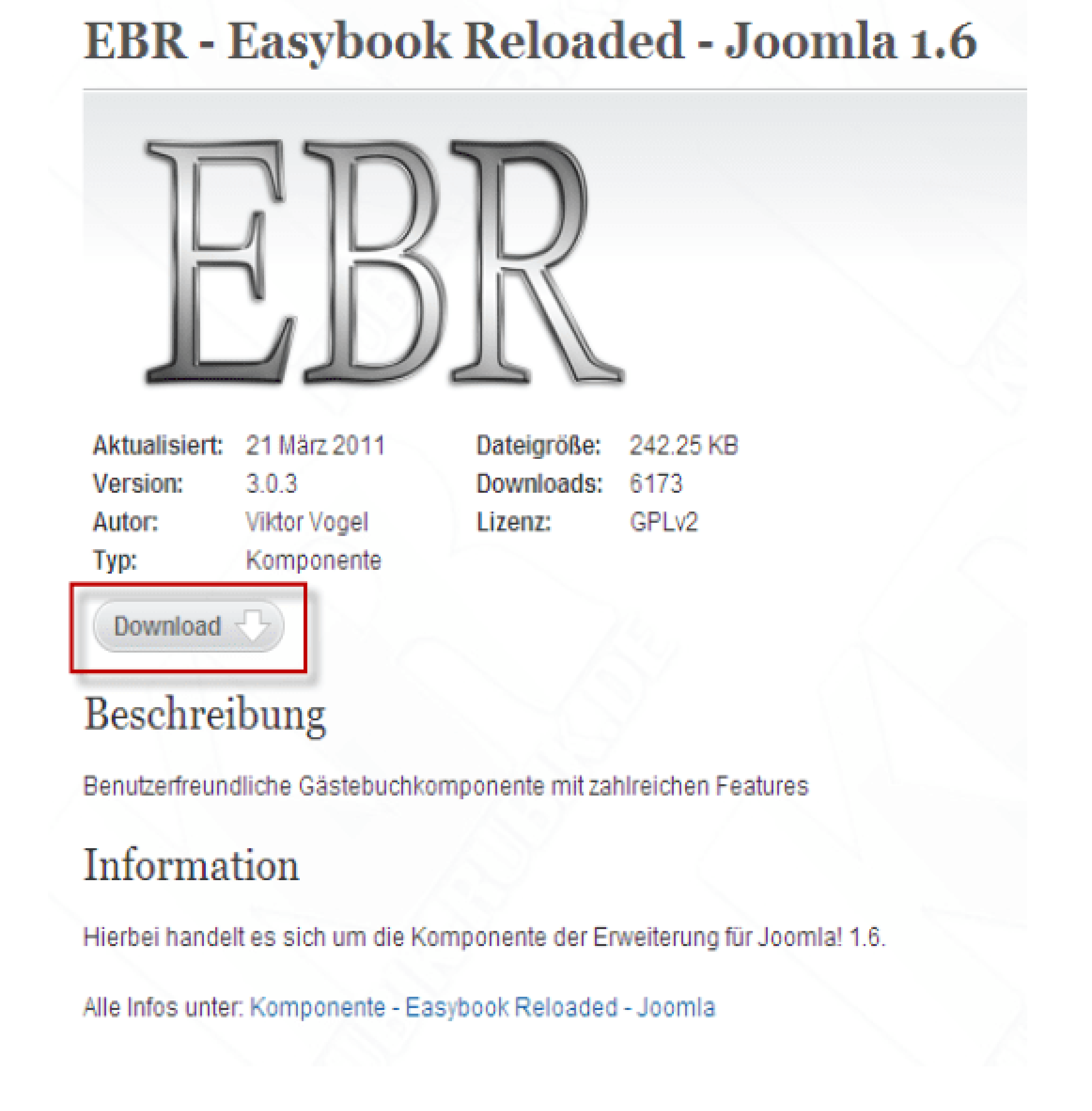


图 22.70　留言板组件下载界面

（1）将该组件下载后保存到本地，然后打开后台管理界面的控制面板，单击“扩展管理”图标。在“安装”选项下选择留言板组件文件，单击“上传并安装”按钮，完成该组件的安装。

（2）在后台管理导航栏的“组件”的下拉菜单中选择留言板组件（Easybook Reloaded），然后单击 Settings 子栏目进入留言板组件的设置界面。

该组件的设置分为 4 个部分，即 Rights 选项、Configuration 选项、Form 选项和 Spam Protection 选项。根据前台页面的展示要求调整 Form 选项的参数设置，如图 22.71 所示。

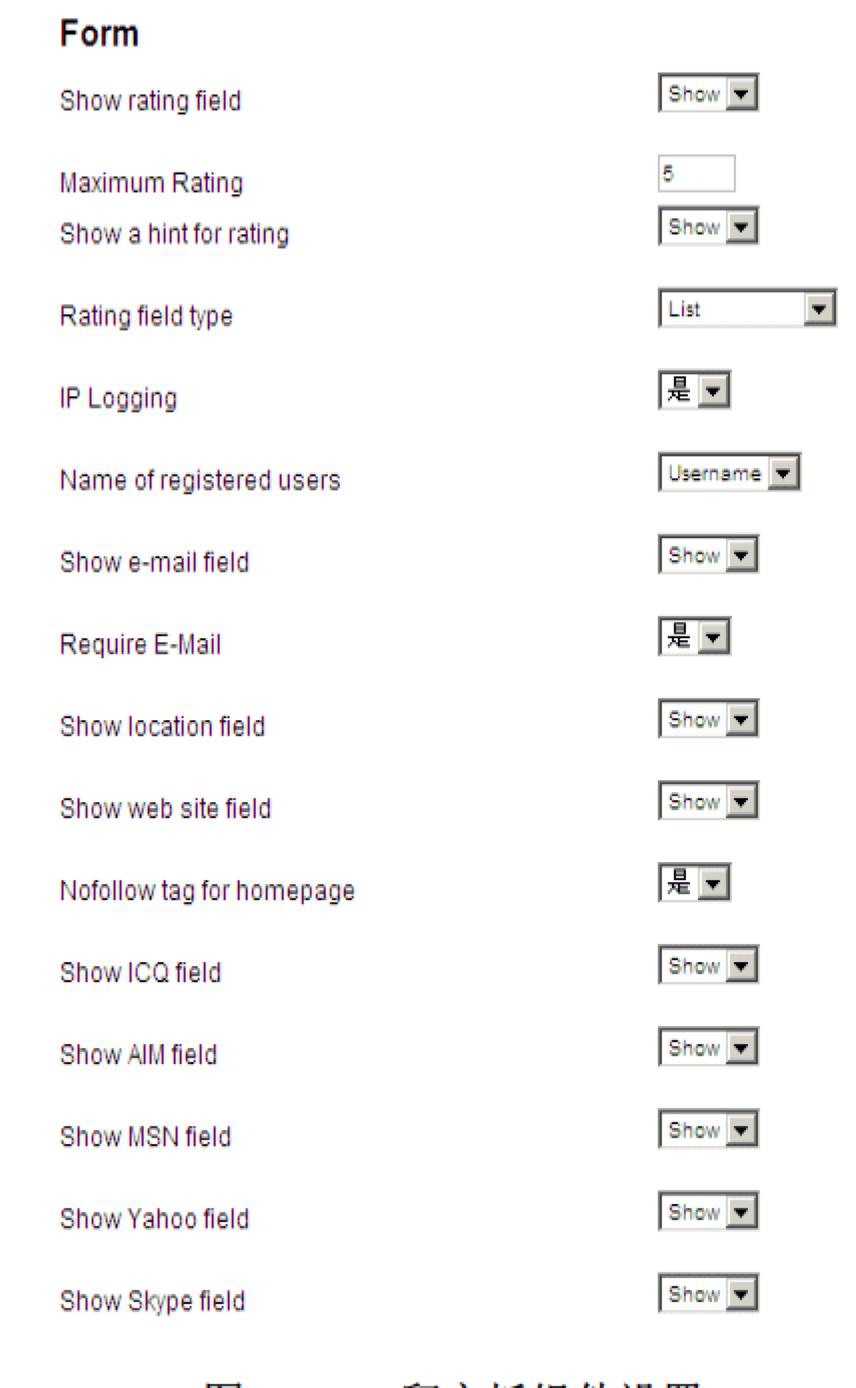


图 22.71　留言板组件设置

（3）在后台管理导航栏的“组件”的下拉菜单中选择留言板组件（Easybook Reloaded），然后单击 manage Entries 子栏目进入留言板组件的留言管理界面，如图 22.72 所示。从图中可以看到已经发表的留言信息，除了可以查看该留言的状态以外，还可以对该留言进行“停止发布”、“删除”等基本管理操作。



图 22.72　留言管理

（4）打开前台的用户留言界面，如图 22.73 所示。该留言界面的核心是一组表单组件构成，包含了留言用户需要填写的必要信息，其中有“*”号标识的选项为必填项目。

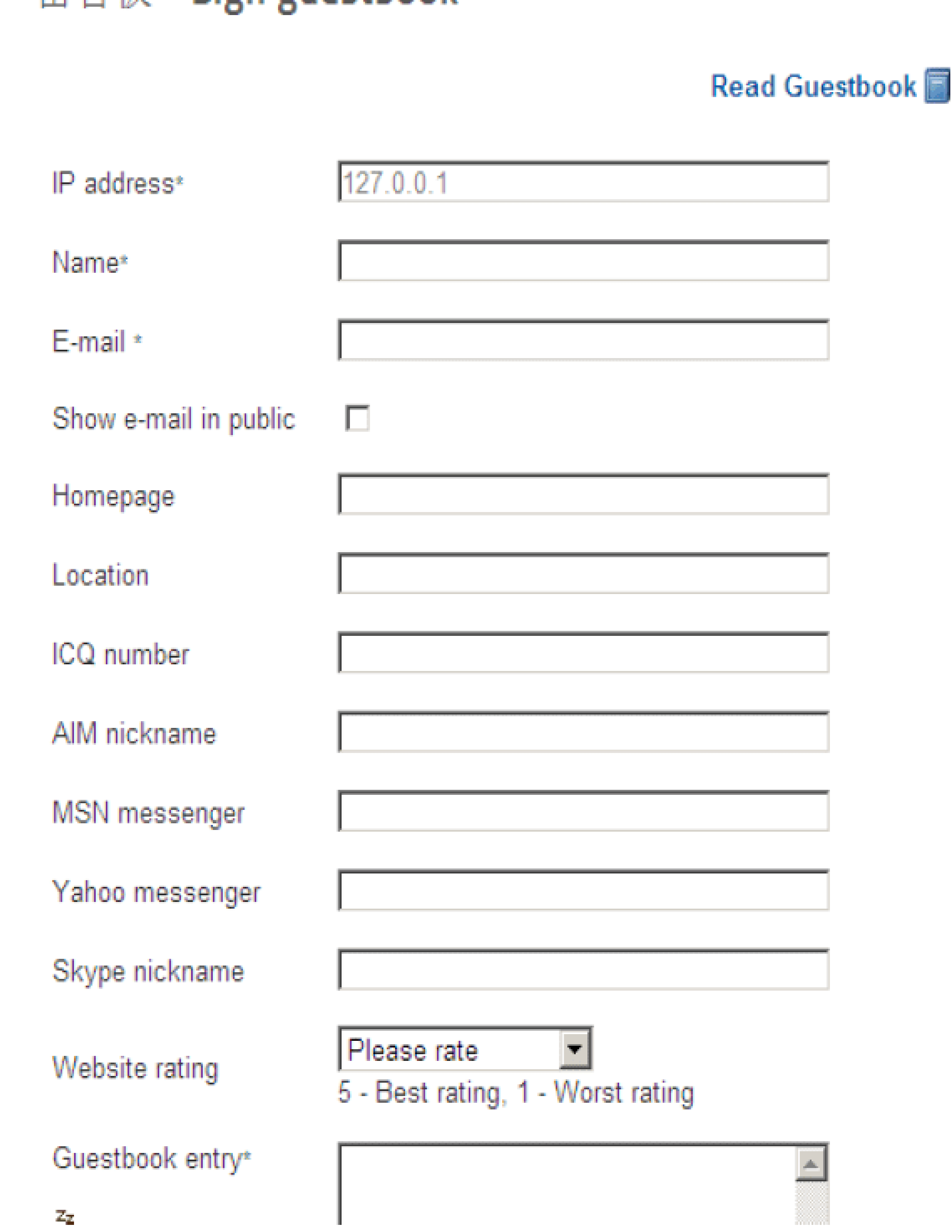


图 22.73　前台留言表单

22.8.5　模板基础及规划

在开始 Joomla 模板的规划之前，先来介绍 Joomla 模板的基础知识。Joomla 模板本身是页面代码文件，它兼容传统的 Table 方式和最新的 Div+Css 的方式。但是通常建议使用 Div+Css 的方式，这将更加符合 W3C 组织的规范适合商业化网站的运营要求，同时有利于搜索引擎收录。

1．模板结构

标准的模板组成分为两个部分，即程序文件和模板文件，对应放在 Joomla 根目录的/templates/目录下。例如，一个文件名为 beez_20 的模板文件对应的目录如下。

```
/templates/ beez_20/
```

在一个模板目录中，必须有关键文件，即模板配置文件（/beez_20/templateDetails.xml）和入口文件（/beez_20/index.php）。而且名称和位置要完全遵守规则，以便 Joomla 系统调用。

（1）模板配置文件（templateDetails.xml）是 XML 格式文件，该文件用来设置基本的安装参数，即加载该模板时需要哪些文件，同时也描述了作者、版权信息等；还有模板使用的图片文件的清单，同时这个文件也是安装文件。下面是一个 XML 格式的模板配置文件（templateDetails.xml）的例子，该文件代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<install version="1.6" type="template">
<name>beez 20</name>
<version>1.0</version>
<creationDate>December 06</creationDate>
<author>Author</author>
<authorEmail>youremail@vip.com</authorEmail>
<authorUrl>www.Joomla.org</authorUrl>
<copyright>GPL</copyright>
<description><![CDATA[three column template]]></description>
<files>
    <filename>index.php</filename>
    <filename>templateDetails.xml</filename>
    <filename>js/somefile.js</filename>
    <filename>images/threecol-1.jpg</filename>
    <filename>images/threecol-2.jpg</filename>
    <filename>css/layout.css</filename>
    <filename>css/template_css.css</filename>
</files>
<params>
    <param  name="showComponent"  type="radio"  default="1"  label="Show
    Component" description="Show/Hide the component output">
    <option value="0">No</option>
    <option value="1">Yes</option>
    </param>
</params>
</install>
```

【代码解读】

```
<install version="1.6" type="template">
```

上述代码表示该文件的版本和类型，其中参数 type="template"表示该安装程序的安装类型的是一个模板。

```
<name>beez_20</name>
```

该标签内容是模板的名称。

注意：因为名称用来创建目录，所以不能有不符合目录要求的字符。

```
<creationDate>December 06</creationDate>
```

该标签内容是模板建立的日期。

```
<author>author</author>
```

该标签内容是模板开发者姓名。

```
<copyright>GPL</copyright>
```

该标签内容是版权信息。

```
<authorEmail>youremail@vip.com</authorEmail>
```

该标签内容是开发人员的邮件地址。

```
<authorUrl>www.joomla.org</authorUrl>
```

该标签内容是作者站点链接。

```
<version>1.0</version>
```

该标签内容是版本信息。

```
<files>…</files>
```

该标签中的内容表示这部分声明模板包括的 PHP 文件和图片文件，每个文件被列出在<filename>与</filename>之间。

（2）入口文件（index.php）是模板文件中最重要的文件，该文件将“通知”系统如何设置页面布局组件和模块的加载位置，该文件的形式始终是 PHP 与“(X)HTML”“混合”在一起。

首先来看页面头部的 DOCTYPE 定义，DOCTYPE 决定了浏览器如何加载页面，以及如何解释 Css，代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo
$this->language; ?>" lang="<?php echo $this->language; ?>" dir="<?php echo
$this->direction; ?>">
```

【代码解读】

<?php echo $this->language; ?>表示读取设置的语言环境。

通过<jdoc:include type="head"/>代码将引入页面的 header 部分。header 部分通常代码如下：

```
<base href="/http://www.maycode.com/" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="robots" content="index, follow" />
<meta name="keywords" content="" />
<meta name="description" content="" /
><title>Joomla! 1.6 template beez 20 </title>
<link href="/index.php?format=feed&type=rss"
_fcksavedurl=""/index.php?format=feed&type=rss"" mce_href="/index.php?
format=feed&type=rss" rel="alternate" type="application/rss+xml"
title="RSS 2.0" />
<link href="/index.php?format=feed&type=atom"
mce_href="/index.php?format=feed&type=atom" rel="alternate"
```

```
type="application/atom+xml" title="Atom 1.0" />
<link                            href="/templates/ja purity/favicon.ico"
mce href="/templates/ja purity/favicon.ico" rel="shortcuticon" type="
image/x-icon" />
```

【代码解读】

这里包含了几个 meta 标签，这些标签的内容是该模板的基本信息。

```
<?php echo $this->template;?>/css/template_css.css
```

上述代码表示引入模板 Css 文件。

通常模板文件还使用其他一些文件，文件位置及含义如下。

- ❑ /beez_20/template_thumbnail.png：当前模板的浏览器截图（通常使用 140×90 像素尺寸）安装模板完毕以后，这张图片会以“缩略图”的形式在后台模板管理器中展示。
- ❑ /beez_20/css/template_css.css：该模板的 Css 文件
- ❑ /beez_20images/logo.png：模板 logo 文件。
- ❑ 下面介绍一个标准的 Joomla 模板，该模板将完成 Joomla 的基本功能并清楚地显示了 Joomla 模板框架的构成。具体代码如下：

```
<html>
<body>
    <?php echo $mainframe->getCfg('sitename');?>
    <jdoc:include type="modules" name="top" style="xhtml" />
    <jdoc:include type="modules" name="left" style="xhtml" />
    <jdoc:include type="component" style="xhtml" />
    <jdoc:include type="modules" name="right" style="xhtml" />
</body>
</html>
```

【代码解读】

以上代码构成了一个基本的 Joomla 框架模板，该模板定义了如下部分：

- ❑ 站点名称；
- ❑ 顶部区域；
- ❑ 左边区域；
- ❑ 正文部分（组件）；
- ❑ 右部区域。

2. 模板文件安装及规划

前面介绍 Joomla 的基本构成及各组成部分的含义。下面将通过 Joomla 后台安装和管理模板并对模板作相应的规划。

打开后台管理界面的控制面板，单击“扩展管理”图标，在“安装”选项下选择待上传的模板文件（本章使用系统默认的模板），单击“上传并安装”按钮，完成模板文件的安装。然后在后台管理界面的控制面板中单击“模板管理”图标进入模板管理界面，如图 22.74 所示。从图中可以看到，Joomla 模板管理中提供了两种模板浏览的方式，即风格方式（列表）和模板方式（缩略图）。

在模板管理列表页面选择 Beez2 - Default 模板，单击链接进入该模板编辑界面，如图

22.75 所示。模板编辑页面分为以下 3 个功能区域。



图 22.74　模板管理



图 22.75　模板编辑界面

- ❑ 详情区域：显示该模板的基本信息，包括风格名称、样式 、默认语言和样式说明。
- ❑ 菜单指派区域：设置该模板指派的模板。
- ❑ 高级选项区域：调整模板的参数，如页面宽度、标题等。

下面通过示例图，来说明默认模板的结构设计，以及各部分对应的位置关系，如图 22.76 所示。



图 22.76　模板结构设计

从图中可以看到，模板的风格模板分为 10 个区域，分别是 top 区域、user 区域、user1 区域、user2 区域、user3 区域、breadcrumb 区域、left 区域（2 列）、right 区域（3 列）。读者可以根据这个结构设计图，合理地分配各个功能栏目的分布，并做合理的内容配置。

22.8.6　创建导航栏及关联菜单

前面几节已经将内容和显示模板都准备好了。下面通过管理后台创建导航栏，以及栏目关联的菜单。通过菜单与内容的关联就能将内容与导航结合在一起，完成整个网站的搭建工作。

（1）打开后台管理界面的导航条，单击“菜单”选项，选择“菜单管理”，然后单击“新建菜单”选项进入新建菜单界面。新建菜单流程有两个项目需要填写“标题”、“菜单类型”和“描述”。如图 22.77 所示，这里只建立“top”导航栏菜单用于前台用户导航，读者在实际应用中可以创建其他扩展菜单类型，如 main 类型、user 类型等。创建菜单完成后单击“保存”图标完成创建操作。



图 22.77　新建菜单

（2）创建菜单后开始新建“菜单项目”，在“菜单管理”界面中单击“新建”图标，进入“菜单项目”界面如图 22.78 所示。



图 22.78　新建菜单项目

（3）根据设计规划新建 7 个菜单项目，菜单名称及参数设置，如图 22.79 所示。创建完“菜单项目”后将“首页”项目设置为首页（选中五角星图标）。



图 22.79　菜单列表

（4）后台设置部分全部结束，在浏览器中输入地址 http://localhost/joomla/，可以看到新建网站，如图 22.80 所示。



图 22.80　网站前台预览

22.9　小　　结

Joomla 是目前被广泛使用的、免费的开源内容管理系统（CMS）。因其具有多种优秀的特性非常适合快速地搭建网站，同时该系统便于扩展，可以保证日后良好的扩展性。

在第 22.1 节从 Joomla 基础知识讲起，然后介绍 Joomla 整体架构。第 22.2 节介绍 Joomla 目录及文件说明，首先介绍总体的架构（根目录说明），而后分别对模板目录、组件目录、模块目录和插件目录进行详细的说明。第 22.3 节是数据库设计部分的内容，介绍了 Joomla

数据库的基础知识、数据库逻辑关系，以及数据库中各个表功能及说明。从第 22.4 节开始进入 Joomla 的实际应用阶段。第 22.4 节介绍了 Joomla 下载与安装的相关，分别是 Joomla 的下载及安装环境需求、文件包解压及配置设置，然后介绍通过 Joomla 的安装向导搭建系统后台。第 22.5 节是对前面搭建的系统后台的具体应用，即后台基础管理部分的内容，该部分主要是介绍系统后台的基础设置，包含权限角色划分、后台概述、语言管理、媒体文件管理和全局设置。系统后台的基础配置完成后就开始向平台中添加内容，因此在第 22.6 节后台内容管理中，分别介绍如何添加内容，包括菜单管理、类别管理和文章管理。众所周知，Joomla 一项迷人的特性就是便于扩展。在之前介绍过 Joomla 的扩展包括 3 种方式，第 22.7 节中分别对插件、组件和模块的扩展开发做了详细的介绍，并带领读者完成了扩展实例。在了解了前面所有的内容后，第 22.8 节将前面的内容作了一次完整地实例化应用。通过快速建站的实例，将前面讲过的内容通过这个实例完整地结合起来，读者既回顾了之前的知识，也对如何快速建站有了直观地认识和了解。